

Proceedings
Image and Vision Computing
New Zealand 2006

Great Barrier Island, New Zealand
27th – 29th November 2006



Proceedings

Image and Vision Computing New Zealand 2006

*Great Barrier Island, New Zealand
27th – 29th November 2006*

Editors: Patrice Delmas, Jason James, and John Morris



The papers appearing in this volume comprise the Proceedings of Image and Vision Computing New Zealand 2005. They reflect the authors' opinions and are published as presented without change, in the interests of timely dissemination. Their inclusion here does not necessarily constitute endorsement by the IVCNZ 2006 organising committee. Abstracting is permitted with acknowledgement of the source.

ISBN 978-0-473-11792-4

ISBN 0-473-11792-4

Copyright © 2006 Image and Vision Computing New Zealand

Individual articles may be photocopied without fee for the purpose of private study or non-commercial teaching. For other copying, reprint or republication please contact the authors directly. Further copies of the Proceedings are available from: Associate Professor John Morris, Department of Computer Science, Tamaki Campus, The University of Auckland, Private Bag 92019, Auckland, New Zealand, e-mail: jmor159@cs.auckland.ac.nz.

Preface

This volume is the proceedings of the 2006 Image and Vision Computing New Zealand Conference (IVCNZ), held at Great Barrier Island, New Zealand on 27-29 November 2006.

Foreword and Welcome

On behalf of the organising committee we would like to welcome all participants to the 2006 Image and Vision Computing New Zealand Conference.

2006 is the 21st year of what has become a truly international conference. While most participants still originate from New Zealand and nearby Australia, this year we also welcome guests from England, France, The Netherlands, Mexico, Taiwan, Hong Kong, Korea, Pakistan, and Japan. This year we received 134 submissions, of which 38 were accepted for oral and 52 for poster presentations. We would like to acknowledge the considerable efforts of the members of the programme committee; they completed nearly 400 reviews in a very short time and ensured a high quality selection of papers being accepted for the conference programme.

While exhibiting 21st century science, this conference is being held at a location without mobile phone coverage or broadband internet access and accessible only via a gravel road or by sea. Great Barrier Island, the location for this year's conference, is sparsely populated, and the largely unspoilt natural environment is free of many pests that are present on mainland New Zealand. Under these conditions, wildlife flourishes and it is a haven for rare birds and lizards.

We would like to thank the members of the organising committee for their assistance in the smooth running of the conference. They are: Heather Armstrong, Robert Carter, Anita Lai, Cecilia Lourdes, Cliff Hawkins, and Georgy Gimel'farb. Thanks to Robert Amor for the support provided by the University of Auckland Computer Science Department. Special thanks are also due to Brendan McCane (IVNZ 2005 organiser) for his help and useful advice.

Finally, we would like to thank all contributing authors and participants for your involvement. We hope you will be able to take some time out from the conference programme to explore and enjoy your unique surroundings.

Patrice Delmas

John Morris

Great Barrier Island, November 2006

Conference Chair

John Morris (Co-Chair)

Patrice Delmas (Co-Chair)

Programme Committee

Donald Bailey (Massey University, NZ)
Andrew Bainbridge-Smith (U. of Canterbury, NZ)
Mohammed Bennamoun (U. of Western Australia)
Phil Bones (University of Canterbury, NZ)
Chris Bowman (Industrial Research Ltd, NZ)
Roger Browne (Massey University, NZ)
Chaur-Chin Chen (National Hsing Hua University, Taiwan)
Chi-Fa Chen (I-Shou University, Taiwan)
Chia-Yen Chen (University of Auckland, NZ)
Chen, Sei-Wang (National Taiwan Normal University)
Hocine Cherifi (University of Bourgogne, France)
Michael Cree (University of Waikato, NZ)
Michel Couprie (EISEE, France)
Patrice Delmas (CITR, NZ)
Bing Du (Massey University, NZ)
Ulrich Eckhardt (U. of Hamburg, Germany)
Chiou-Shann Fuh (National Taiwan University)
André Gagalowicz (INRIA, France)
Georgy Gimel'farb (University of Auckland, NZ)
Hideaki Goto (Tohoku University, Japan)
Peter Gough (University of Canterbury, NZ)
Richard Green (University of Canterbury, NZ)
Michael Hayes (University of Canterbury, NZ)
Volker Hilsenstein (CSIRO Mathematical and Information Sciences, Australia)
Eunjung Holden (University of Western Australia)
Fay Huang (CSIE Institute, Taiwan)
Atsushi Imiya (University of Chiba, Japan)
Herbert Jahn (Deutsches Zentrum fuer Luft- und Raumfahrt, Germany)
Jason James (University of Auckland, NZ)
Yongkyu Kim (South Korea)
Scott King (Texas A&M University-Corpus Christi, USA)
Reinhard Klette (CITR, NZ)
Richard Lane (Applied Research Associates NZ Ltd, NZ)
Philippe Leclercq (France Telecom R&D, France)
Wen-Nung Lie (NCCU, Taiwan)
Frank Luthon (CS Lab. LIUPPA, France)
Anthony Maeder (e-Health Research Centre / CSIRO, Australia)
Jorge Marquez (Universidad Nacional Autonoma de Mexico)
Brendan McCane (University of Otago, NZ)
Stephen McNeill (Landcare Research, NZ)
Rick Millane (University of Canterbury, NZ)
John Morris (CITR, NZ)
Heather North (Landcare Research, NZ)

David Pairman (Landcare Research, NZ)
David Penman (IRL, NZ)
Edwige Pissaloux (LRP/CEA, France)
Amal Punchihewa (Massey University, NZ)
Ralf Reulke (Deutsches Zentrum für Luft-und Raumfahrt, Germany)
Bodo Rosenhahn (Max-Planck-Centre, Germany)
Johann Schoonees (IRL, NZ)
David Squire (Monash University, Australia)
Antonio Torralba (MIT, USA)
Robert Valkenburg (IRL, NZ)
Tiangong Wei (CITR NZ)
Peter Whigham (University of Otago, NZ)
Alexander Woodward (CITR NZ)
Burkhard Wuensche (University of Auckland, NZ)
Mengjie Zhang (Victoria University of Wellington, NZ)

Organising Committee

Scientific: Georgy Gimel'farb

Editing: Jason James

Financial: Anita Lai

Admin: Cecilia Lourdes, Heather Armstrong

Technical: Cliff Hawkins, Robert Carter

Sponsors

Communication and Information
Technology Research (CITR)

Control Vision

Hoare Research Software Ltd (HRS)

Savant Information Systems

Cover Image Credits

Front Cover

Kaitoke Beach, Great Barrier Island, New Zealand.

Photograph courtesy of Celine Duwig.

Title Page

Reconstruction of corrupted images using principal component analysis.

Reference: Joint Outliers and Principal Component Analysis *Georgy Gimel'farb, Alexander Shorin, and Patrice Delmas*

Back Cover

Top row

3D head and interactively styled hair models.

Reference: Interactive Styling of Virtual Hair *Rui Zhang and Burkhard Wünsche*

Middle Row (from left to right)

Kauri Dams on the Kaiaraara Track, Great Barrier Island.

Bridge on the Kaitoke Hot Springs Track, Great Barrier Island.

Photographs courtesy of Patrice Delmas.

Bottom Row

Stumpy the gnome and a range image generated by the University of Waikato Range Imager.

Reference: The Waikato Range Imager *M. J. Cree, A. A. Dorrington, R. M. Conroy, A. D. Payne, and D. A. Carnegie*

Table of Contents

Monday 27th November

Keynote Speaker:

9:00 – 9:40

Posters:

9:40 – 11:00

Signal / Image Processing

Parameter Analysis for Mixture of Gaussians Model	1
<i>Qi Zang and Reinhard Klette</i>	
Performance Evaluation of Accurate Ellipse Fitting	7
<i>Kenichi Kanatani</i>	
Rectifying Images for Stereo Vision	13
<i>Y. Lin, A. Woodward, D. An, J. Morris, P. Delmas, and G. Gimel'farb</i>	
An Image Data Hiding Scheme being Perfectly Imperceptible to Histogram Attacks.....	19
<i>Hung-Min Sun, Yao-Hsin Chen, and King-Hang Wang</i>	
Chromatic Variance Prediction.....	25
<i>Robert N. Grant and Richard D. Green</i>	
Near optimal non-uniform interpolation for image super-resolution from multiple Images.....	31
<i>A. Gilman and D.G. Bailey</i>	
Storing and Accessing Large Images using Summed Area Tables.....	37
<i>Volker Hilsenstein</i>	
A comparison of noise in CCD and CMOS image sensors.....	43
<i>K. Irie, A. E. McKinnon, K. Unsworth, and I. M. Woodhead</i>	
Pros and Cons of the Nonlinear LUX Color Transform for Wireless Transmission with Motion JPEG2000.....	49
<i>T. Totozafiny, F. Luthon, and O. Patrouix</i>	
Moment-based Local Descriptor using Scale Invariant Keypoints.....	55
<i>Jae-Sun Han, Gwang-Gook Lee, and Whoi-Yul Kim</i>	
A Hybrid Approach to Man-Made Structure Extraction from Natural Scenes	61
<i>Hang Zhou, David Suter, and Konrad Schindler</i>	
Accelerating calibrated stereo correspondence through concurrent processing	67
<i>Nathan Adams and Richard Green</i>	
Local Texture Patches for Active Appearance Models	73
<i>N. Faggian, A. P. Paplinski, and J. Sherrah</i>	

Oral:

11:00 – 13:00

Signal / Image Processing

Modified Kalman Filtering for Image Super-Resolution	79
<i>C. Newland, D. Gray, and D. Gibbins</i>	

Affine Normalized Contour Invariants using Independent Component Analysis and Dyadic Wavelet Transform	85
<i>Asad Ali and S. A. M. Gilani</i>	
VQ-Based Data Hiding in Images by Minimum Spanning Tree	91
<i>Hung-Min Sun, King-Hang Wang, Hou-Wen Wang, and Chia-Yen Chen</i>	
Morphology-based Stable Salient Regions Detector	97
<i>E. Rangelova and E. J. Pauwels</i>	
A study of 3rd and 4th order Tikhonov smoothing term influence on the convergence of active contours.....	103
<i>Moqing Zhang and Patrice Delmas</i>	
Iterative Target Calibration Using Conformal Geometric Algebra.....	109
<i>Robert J. Valkenburg, Nawar S. Alwesh, Yilan Zhao, and Reinhard Klette</i>	

**Oral:
13:40 – 15:20**

Security

Fingerprint Matching using Enhanced Shape Context.....	115
<i>Paul W.H. Kwan, Junbin Gao, and Yi Guo</i>	
Towards real time difference imaging in the far blue (390-440 nm).....	121
<i>G. M. Miskelly, and J. H. Wagner</i>	
Watermarking on 3D Model.....	127
<i>Chia-Yen Chen and Chi-Fa Chen</i>	
License Plate Detection and Classification using a Space Displacement Neural Network	133
<i>M. Johnson, A. Barczak, and S. Russell</i>	
Multiscale Contrast Patterns for Image Tamper Detection	137
<i>M. K. Bashar, N. Ohnishi, H. Kudo, T. Matsumoto, and Y. Takeuchi</i>	

**Posters:
15:20 – 16:40**

Visualisation and Graphics

Interactive Styling of Virtual Hair	143
<i>Rui Zhang and Burkhard Wünsche</i>	
Classification of 3D LIDAR Point Clouds for Urban Modelling.....	149
<i>E. H. Lim and D. Suter</i>	
Real-Time Interaction Techniques for Meshless Deformation Based on Shape Matching.....	155
<i>Alex Henriques and Burkhard Wünsche</i>	
Terrain Reconstruction using LADAR and Optical Sensor Data from an Unmanned Air Vehicle.....	161
<i>D. Gibbins, L. Swierkowski, P. Roberts, and A. Finn</i>	
Image processing of cryo-electron micrographs of helical crystals - 3D architecture of a novel bacterial appendage	167
<i>J. Li, S. Manning, S. Turner, M. Kikkawa, and A. K. Mitra</i>	
Public Interactive Display Using Front-projection and Infrared-pass Filter Camera ..	173
<i>Cheng-Tse Chu, Dandi Duan, and Richard Green</i>	

Simulation of multi-polarisation SAR imagery	179
<i>S. J. McNeill, D. Pairman, H. C. North, S. E. Belliss</i>	
Extracting Surface Curvature from Noisy Scan Data	185
<i>J. Rugis</i>	
Occlusion Removal in Image for 3D Urban Modelling	191
<i>E. H. Lim and D. Suter</i>	
Modelling Interactions with a Computer Representation of the Upper Gastrointestinal System	197
<i>Gastélum Alfonso and Márquez Jorge</i>	
Acquiring Visual Hulls by Voxels	203
<i>Yu-xuan HONG and Richard Green</i>	
Simulation of Medical Imaging Modalities - A Tool for Numerical Evaluation of Image Processing Algorithms.....	209
<i>F. Uhlemann</i>	
Analysis of Differential Interference Contrast Microscopy Images of the Retina	215
<i>D. H. Wojtas, B. Wu, P. Wenig, P. K. Ahnelt, P. J. Bones, and R. P. Millane</i>	
ROBPCA-SIFT: a feature point extraction method for the consistent with epipolar geometry in endoscopic images.....	221
<i>J. S. Oh, H. C. Kim, J. M. Koo, J. S. Yu, T. H. Kang, J. D. Lee, and M. G. Kim</i>	
Towards nuclear phenotype recognition in single channel fluorescence microscopy images.....	227
<i>I. Sintorn, L. Bischof, R. Lagerstrom, M. Buckley, and A. Hoffman</i>	
Oral:	
16:40 – 18:00	
3D	
The Waikato Range Imager	233
<i>M. J. Cree, A. A. Dorrington, R. M. Conroy, A. D. Payne, and D. A. Carnegie</i>	
Digital Speckle Photogrammetry	239
<i>Yizhe Lin, John Morris, Quentin Govignon, and Simon Bickerton</i>	
Interactive Hand-held 3D Scanning	245
<i>R. J. Valkenburg, D. W. Penman, J. A. Schoonees, N. S. Alwesh, and G. T. Palmer</i>	
3D Visualisation Techniques for Multi-Layer Display™ Technology.....	251
<i>Vijay Prema, Gary Roberts, and Burkhard Wünsche</i>	
SRICP: An Algorithm for Matching Semi-Rigid Three-Dimensional Surfaces	257
<i>Ajmal Mian, Mohammed Bennamoun, and Robyn Owens</i>	

Tuesday 28th November

Keynote Speaker:

9:00 – 9:40

Posters:

9:40 – 11:00

Applications

Objective Colour Measurement of Tomatoes and Limes	263
<i>H. M. W. Bunnik, D. G. Bailey, and A. J. Mawson</i>	

Athlete Performance Video Overlay	269
<i>S. Sarjeant and R. Green</i>	
Image Processing of Meat Images for Visible/Near Infrared Spectroscopy Reference	275
<i>Lee Streeter, G. Robert Burling-Claridge, and Michael J. Cree</i>	
Quality Assessment of Retinal Images	281
<i>Y. Kwon, A. Bainbridge-Smith, and A. B. Morris</i>	
Results of a multiple-baseline interferometric synthetic aperture sonar in shallow Water	287
<i>M. P. Hayes</i>	
Monocular tracking of swimmers from a stationary viewpoint.....	293
<i>C. P. Huynh and R. Green</i>	
Accounting for User Familiarity in User Interfaces	299
<i>C. A. D'H Gough, R. Green, and M. Billinghurst</i>	
Image Denoising Using a New Line-Field	305
<i>Ngoc-Thuy Le and Kah-Bin Lim</i>	
Augmenting Sports Grounds with Advertisement Replacement	311
<i>D. K. Barrow and R. Green</i>	
A Hybrid Approach for Tracking Eye Pupils	319
<i>M. Schoo and R. Green</i>	

**Oral:
11:00 – 13:00**

Bio-Medical Imaging

Ultrasound Image Segmentation With Multilayer Perceptron-Based Level Sets.....	325
<i>M. Mora, C. Tauber, and H. Batatia</i>	
An Automated System for Microscopy Imaging and Analysis of Histology Slides with an Application in Sheep Meat Morphometry	331
<i>V. Hilsenstein, P. Jackway, and P. Allingham</i>	
Morphological Averaging of Anatomical Shapes Using Three-Dimensional Distance Transforms	337
<i>Márquez Jorge, Patrice Delmas, Isabelle Bloch, and Francis Schmitt</i>	
Image Analysis and Modelling of Disorder in the Myosin Lattice of Vertebrate Muscle	343
<i>C. H. Yoon, N. D. Blakeley, A. Goyal, and R. P. Millane</i>	
Vision based Human Activity Detection for Eldercare and Security.....	349
<i>Nigel Pereira, Liyanage C. De Silva, and Amal Punchihewa</i>	
Automatic Recognition of Light-Microscope Pollen Images	355
<i>G. P. Allen, R. M. Hodgson, S. R. Marsland, G. Arnold, R. C. Flemmer, J. Flenley, and D. W. Fountain</i>	

Wednesday 29th November

Keynote Speaker:

9:00 – 9:40

Posters:

9:40 – 11:00

Recognition and Detection

Tracking Articulated Objects using Improved Particle Filters.....	361
<i>Martin Tosas and Li Bai</i>	
Detection and Removal of Global and Local Noise in Realtime Video Streams.....	367
<i>A. Clark and R. Green</i>	
Matching Moving Objects by Parts with a Maximum Likelihood Criterion.....	373
<i>Eric Dahai Cheng and Massimo Piccardi</i>	
Semi-supervised Silhouette Detection for Thermal Imaging.....	379
<i>Surya Prakash and Antonio Robles-Kelly</i>	
A simple and efficient eye detection method in color images	385
<i>D. Sidibe, P. Montesinos, and S. Janaqi</i>	
Access Control with Session Based Face Tracking.....	391
<i>Amadeus Rainbow and Richard Green</i>	
A New Rapid Feature Extraction Method for Computer Vision based on Moments ..	395
<i>A. L. C. Barczak and M. J. Johnson</i>	
A Robust Efficient Motion Segmentation Algorithm	401
<i>Hongzhi Gao and Richard Green</i>	
Camera Egomotion Tracking using Markers.....	407
<i>Brendon Kelly and Richard Green</i>	
Fast and Adaptive Block-based Motion Estimation for Video Coding.....	413
<i>G. Sorwar and M. Murshed</i>	
A Simple Model-Free Approach to Posture Recognition	419
<i>R. Raghavan, K. C. Aw, S. Xie</i>	
Genetic Programming for Object Detection	425
<i>Mengjie Zhang, Urvesh Bhowan, and Bunna Ny</i>	
Object Indexing and Recognition	431
<i>F. Souami and S. Aouat</i>	

Oral:

11:00 – 13:00

Motion / Image Processing

Detection of Cirrus Streak Utilizing Cloud Shape and Movement.....	437
<i>H. Ikeda, R. Saegusa, and S. Hashimoto</i>	
Region-based MRF Model for Moving Object Segmentation.....	443
<i>S. K. Hwang and W. Y. Kim</i>	
Structured Combination of Particle Filter and Kernel Mean Shift Tracking.....	449
<i>A. Naeem, S. Mills, and T. Pridmore</i>	

Image Segmentation Using an Active Contour Model	455
<i>Byeong Rae Lee, YongKyu Kim, and Hyunchul Kang</i>	
Joint Outliers and Principal Component Analysis	461
<i>Georgy Gimel'farb, Alexander Shorin, and Patrice Delmas</i>	
Integrated Test Pattern Generator and Measurement Algorithm for Colour Compression Artefacts in Ubiquitous Colour Spaces	467
<i>G. A. D. Punchihewa, D. G. Bailey, and R. M. Hodgson</i>	

Oral:
13:40 – 15:20

Stereo

3D Reconstruction from an Uncalibrated Long Image Sequence.....	473
<i>T. Osawa, I. Miyagawa, K. Wakabayashi, K. Arakawa, and T. Yasuno</i>	
Stereo Vision: Concurrent Matching vs Optimisation.....	479
<i>Georgy Gimel'farb, John Morris, Patrice Delmas, and Jiang Liu</i>	
Image Intensifier Characterisation	487
<i>A. D. Payne, A. A. Dorrington, M. J. Cree, and D. A. Carnegie</i>	
Noise Models for Symmetric Dynamic Programming Stereo.....	493
<i>Zhen Zhou, Georgy Gimel'farb, and John Morris</i>	

Oral:
15:20 – 17:00

Applications

Tracking iris surface deformation using Elastic Graph Matching.....	499
<i>Sammy S. S. Phang, Wageeh Boles, and Michael J. Collins</i>	
Perceptually Correct Image Space Soft Shadows	505
<i>R. Rountree, R. Rayudu and D. Brebner</i>	
Hardware implementation of the Maximum Subarray Algorithm for Centroid Estimation.....	511
<i>S. J. Weddell and B. N. Langford</i>	
A study on GPU implementation of March's regularization method for optical flow computation	517
<i>Yoshiki Mizukami and Katsumi Tadamura</i>	
Determination of Average Wind Velocity using Generalised SCIDAR.....	523
<i>J. L. Mohr, R. A. Johnston, C. C. Worley, and P. L. Cottrell</i>	

Parameter Analysis for Mixture of Gaussians Model

Qi Zang and Reinhard Klette

Department of Computer Science, Tamaki Campus, The University of Auckland
Auckland, New Zealand

Email: qzan001@ec.auckland.ac.nz

Abstract

Background subtraction is one of the main techniques to extract moving objects from background scenes. A mixture of Gaussians is a common model for background subtraction. There are several parameters involved in such a model. Obviously, the assignment of initial values to these parameters affects the accuracy of background subtraction. In this paper, we analyze in detail the impact of different initial parameter values based on our model implementation. Both indoor and outdoor video sequences have been tested. This parameter value analysis provides suggestions how to choose suitable initial parameter values, assign reasonable thresholds which ensure better results, while using a mixture of Gaussians model in video surveillance applications.

Keywords: mixture of Gaussians model, parameter analysis, video surveillance

1 Introduction

The mixture of Gaussians model (MOGS) became increasingly popular in image sequence analysis due to its robustness and stability [3][4]:

1. *MOGS characterize static scenes.*

A common example is the paper [9] by Stauffer and Grimson which models each background pixel's distribution using a mixture of Gaussians model; this model allowed (for example) to monitor continuously a university campus. It learns patterns of activities at a given site, then monitors and classifies activities based on these learned patterns. The system provides statistical descriptions of typical activity patterns despite of rainy, snowy, or sunny weather.

2. *MOGS characterize object colors or object trajectories.*

For examples of applications of mixture of Gaussians model for modelling object colors or tracking of a moving object, see papers [6, 7] by Raja et al. Gaussians mixture models were used to estimate probability densities of the color of human skin, clothing, and background. These models were used to detect, track, and segment people, faces, or hands [8].

Further mixture of Gaussians model applications are to model noise distributions or shaded areas [2]. Paper [2] presents a method for detecting moving object shadows against a static background scene using a Gaussian shadow model. The chosen shadow model is parameterized with several features including the orientation, mean and center position of a shadow region. Using

a mixture of Gaussians model to characterize moving objects also allows to deal with partial occlusions (but often in a time-consuming way).

In this paper we use a mixture of Gaussians model for modelling static background scenes. We present our results of implementing a mixture of Gaussians model based on both indoor and outdoor video sequences. A detailed analysis of assigning different values to the parameters in a mixture of Gaussians model is presented. These experimental results provide some guidelines for the selection of different parameter values.

2 Related work

An important property of Gaussian distributions is that they still remain Gaussian distributions after any linear transformation. This property is one of the reasons that the Gaussian models are very commonly used for solving estimation problems [1]. Gaussian models are widely used in adaptive systems. Especially in video surveillance applications, normally a Gaussian distribution is assumed in order to make the system adaptive to uncontrolled changes like in illumination, outdoor weather, color changes, and so on.

A Gaussian mixture is a *pdf* (i.e., point distribution function) consisting of a weighted sum of Gaussian densities [1]. The Gaussian mixture model belongs to a class of density models which combine several functions as additive components.

Let \mathbf{X}_t be the variable which represents the current pixel in frame \mathbf{I}_t , K is the number of distributions, and t represents time (i.e., the frame index), $\omega_{i,t}$ is an estimate of the weight of the i th Gaussian in the mixture at time t , η is a Gaussian probability density function, $\mu_{i,t}$

is the mean value of the i th Gaussian in the mixture at time t , $\Sigma_{i,t}$ is the covariance matrix of the i th Gaussian in the mixture at time t . These functions are combined together to provide a combined density function, which can be employed, for example, to model colors of a dynamic scene or object. Probabilities are computed for each color pixel while a model is constructed.

A Gaussian mixture model can be formulated in general as follows:

$$P(\mathbf{X}_t) = \sum_{i=1}^K \omega_{i,t} \eta(\mathbf{X}_t; \mu_{i,t}, \Sigma_{i,t}) \quad (1)$$

where, obviously,

$$\sum_{i=1}^K \omega_{i,t} = 1 \quad (2)$$

The mean of such a mixture equals

$$\mu_t = \sum_{i=1}^K \omega_{i,t} \mu_{i,t} \quad (3)$$

that is, the weighted sum of the means of the component densities.

For example, papers [5, 6, 7] are all based on using the Gaussian mixture model. In [7], a number of Gaussian functions are taken as an approximation of a multi-model distribution in color space, and conditional probabilities are computed for all color pixels, probability densities are estimated from the background colors, and peoples' clothing, heads, hands, and so forth. Two assumptions are made, one is that a person of interest in an image will form a spatially contiguous region in the image plane. Another is that the set of colors for either the person or the background are relatively distinct, the pixels belonging to the person may be treated as a statistical distribution in the image plane.

An adaptive technique based on the Gaussian mixture model is discussed in [9] for the tracker module of a video surveillance system. This technique is to model each background pixel as a mixture of Gaussians. The Gaussians are evaluated using a simple heuristics to hypothesize which are most likely to be part of the "background process". Each pixel is modelled by a mixture of K Gaussians as stated in Equation (1), where K is the number of distributions. Normally, K equals 3, 4 or 5 in practice. Every new pixel value \mathbf{X}_t is checked against the existing K Gaussian distributions until a match is found. Based on the matching results, the background is updated as follows:

\mathbf{X}_t matches component i if \mathbf{X}_t is within 2.5 standard deviation of this distribution (multiple matches are possible); in case of such a match, the parameters of the i th component are updated as follows:

$$\omega_{i,t} = (1 - \alpha)\omega_{i,t-1} + \alpha \quad (4)$$

$$\mu_{i,t} = (1 - \rho)\mu_{i,t-1} + \rho\mathbf{X}_t \quad (5)$$

$$\sigma_{i,t}^2 = (1 - \rho)\sigma_{i,t-1}^2 + \rho(\mathbf{X}_t - \mu_{i,t})^\top (\mathbf{X}_t - \mu_{i,t}) \quad (6)$$

where $\rho = \alpha P(\mathbf{X}_t | \mu_{i,t-1}, \Sigma_{i,t-1})$. α is the predefined learning parameter, $\sigma_{i,t}^2$ is the variance of the i th Gaussian in the mixture at time t , μ_t is the mean of the pixel at time t , \mathbf{X}_t is (as above) the recent pixel at time t .

The parameters for all the unmatched distributions remain unchanged, what means that

$$\mu_{i,t} = \mu_{i,t-1} \quad \text{and} \quad (7)$$

$$\sigma_{i,t}^2 = \sigma_{i,t-1}^2 \quad (8)$$

But the corresponding weights $\omega_{i,t}$ need to be adjusted using the formula:

$$\omega_{i,t} = (1 - \alpha)\omega_{i,t-1} \quad (9)$$

If \mathbf{X}_t matches none of the K distributions, then the least probable distribution (i.e., the distribution with the lowest weight) is replaced by a distribution where the current value acts as its mean value, the variance is chosen to be "high" and the a-priori weight is "low" [9].

The background estimation problem is solved by specifying the Gaussian distributions, which have the most supporting evidence and the least variance. Because the moving object has larger variance than a background pixel, so in order to represent background processes, first the Gaussians are ordered by the value of $\omega_{i,t} / \|\Sigma_{i,t}\|$ in decreasing order. The background distribution stays on top with the lowest variance by applying a threshold T , where

$$B = \operatorname{argmin}_b \left(\frac{\sum_{i=1}^b \omega_{i,t}}{\sum_{i=1}^K \omega_{i,t}} > T \right) \quad (10)$$

(Note that the denominator is supposed to be equal to 1 in case of proper normalization.) All pixels \mathbf{X}_t which do not match any of these components will be marked as foreground.

3 Analysis of parameter values

Threshold T is to define the fraction between background distribution and foreground distribution. This value is based on the background scene and the number of components in the Gaussian mixture model. We can obtain it from a testing procedure before starting the real application system. A small value of T (say, $T = 0.1$) will lead to a situation, in which not all background distribution is covered; a large T value (say, $T = 0.9$) will lead to a situation in which the foreground distribution is "merging" with the background distribution. The T value we used in our program equals 0.79. We will analyze other parameter values in the following.

3.1 Number of components

K denotes the number of components in a Gaussian mixture model. For simple indoor scenes, a small value of K is sufficient, perhaps $K = 2$; for outdoor complex scenes, a larger K is needed, usually 3, 4, or 5.

Figure 1 presents our indoor testing results without removing noise. The values we assigned to K are from 1 to 5. Figure 1 illustrates our general experience that adding more components in a Gaussian mixture model does not help in improving the quality of the extracted foreground region. On the contrary, the quality of the extracted foreground region even decreased for $K > 1$. This is because although more components can model more distributions, indoor simple scenes are often not characterized by complex changes, and updating components of the model causes more noise. Figure 1 illustrates that $K = 1$ or $K = 2$ appears here to be the best choice.

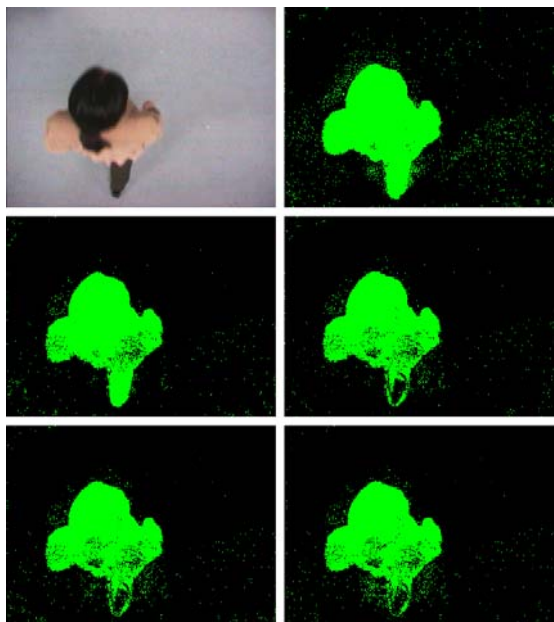


Figure 1: Top left: an original image of a captured sequence. Top right: result for $K = 1$. Middle left: $K = 2$. Middle right: $K = 3$. Bottom left: $K = 4$. Bottom right: $K = 5$.

In complex outdoor scenes, assigning $K = 1$ or $K = 2$ is typically insufficient. For example, we also tested on a winter traffic sequence (uncommon to Auckland) which involves bad weather, snow, and wind. In order to control the movement of snow, waving leaves, and so forth, we defined pixels with values within 4 times standard deviation to be background. K is set to 3. Figure 2 illustrates that, although most small movement of tree leaves and snow are controlled, foreground regions of walking people are missing. The extracted foreground regions are not clear, because vehicles are not running as fast as they normally would on a highway without

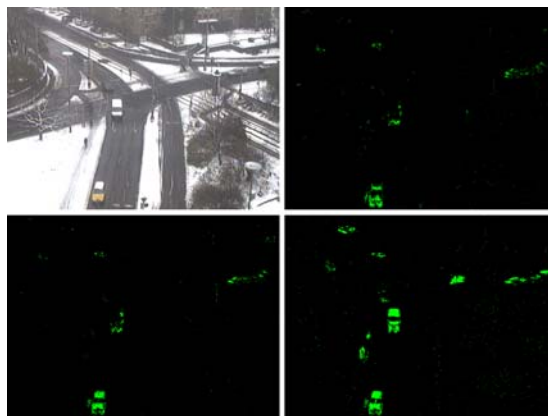


Figure 2: Top left: an original image of the sequence. Top right: result for $K = 3$. Bottom left: $K = 4$. Bottom right: $K = 5$.

snow. We increased the value of K to 4 and 5. The quality of the extracted regions improved.

3.2 Learning rate α

There are two learning rates defined in [9]: one is the predefined learning rate α , the other is the calculated learning rate ρ . ρ is used as a second filter in [9]. As we already summarized in [10], using ρ as a second learning rate is not helpful. We tried using ρ with a very small value, say, less than 10^{-5} . The increase in computation time is costly. In general, assume that the computation time of using one learning rate α is m seconds; then the computation time of using two learning rates α and ρ was greater than $2m$ seconds.

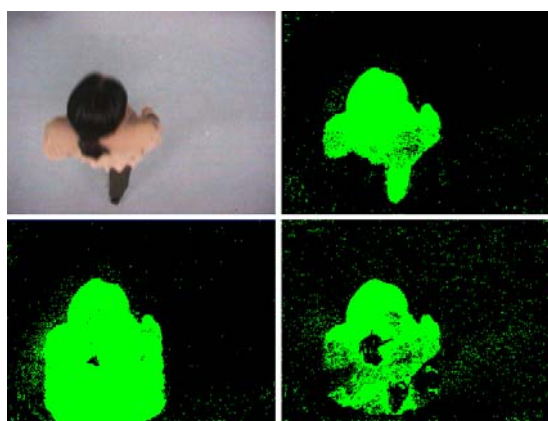


Figure 3: Top left: an original image of the sequence. Top right: result for $\alpha = 0.1$. Bottom left: $\alpha = 0.01$. Bottom right: $\alpha = 0.5$.

In conclusion, we used one learning rate α only. How to assign a reasonable value to α will depend on the given background scenery. A slowly changing background scene needs a small learning rate, a fast chang-

ing background scene needs a larger learning rate. The value α can be obtained from a testing sequence. Here we present an example of using different α values for indoor testing data, see Figure 3. The results in Figure 3 are background estimation before removing noise. Figure 3 illustrates that using value $\alpha = 0.1$ is the best choice for the illustrated cases.

3.3 Assigning initial values

There is an initialization procedure when starting the surveillance system. Assigning different initial values in this procedure will affect the extraction of foreground regions. There are two values that need initial consideration: mean and standard deviation. We will discuss them separately.

Regarding the mean value, from our testing sequences we conclude that assigning either a very large value or a very small value can be considered to be of benefit. Figure 4 shows test results without removing noise. Increasing the mean value from zero to 50 does not impact the extraction of the walking person (as foreground region) very much, and this was experienced for various scenes. In the shown example, the result improved for value 100, but this is not standard for complex backgrounds, and results often were less satisfactory for mean around 100, compared to means below 50. (There are possibilities that the foreground region will be misclassified as the background region.) Large mean values, such as 355 or -999, also proved to be more robust.

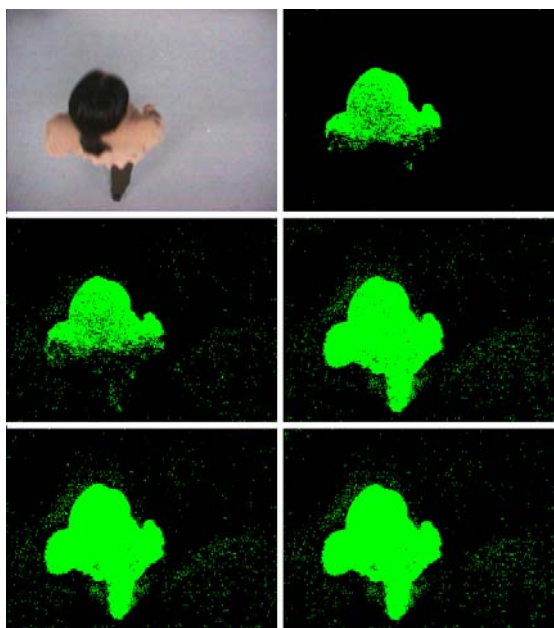


Figure 4: Top left: an original image of the sequence. Top right: result for mean = 0. Middle left: mean = 50. Middle right: mean = 100. Bottom left: mean = 355. Bottom right: mean = -999.

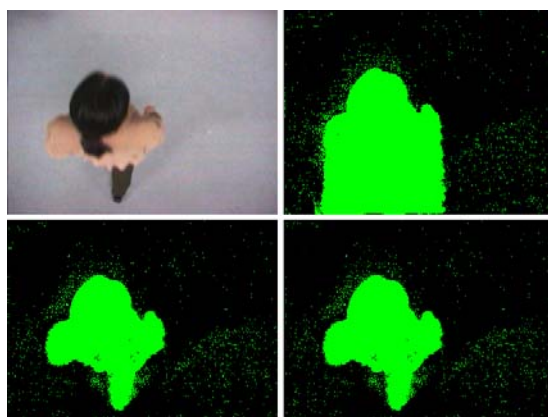


Figure 5: Top left : an original image of the sequence. Top right: result for standard deviation = 0. Bottom left: standard deviation = 100. Bottom right: standard deviation = 350.

In the initialization procedure, we assign in general a very large value to the standard deviation based on our experiments. Figure 5 shows testing results again for the standard sequence used in this paper (without removing noise): for standard deviation equals zero (as an extreme value), many background pixels are misclassified as foreground region even for this simple background. Standard deviation values between 100 and 350 are recommended. In general, using a small value of the standard deviation causes that background pixels are too often classified as foreground distribution.

There are other options to assign a value to the standard deviation. The least probable distribution will be replaced if the current pixel does not match with any of the existing distributions. The mean value will be replaced using the current pixel value. The standard deviation value needs to be large. Figure 6 shows test results without removing noise. If assigning the standard deviation value to 2, then almost the whole scene is classified as being foreground. This is because pixels with lower values of the standard deviation will be easily classified into the foreground distribution. The middle row of Figure 6 are results of assigning standard deviation values to 12 and 42, respectively. The extracted foreground regions improve in these cases. If assigning standard deviation values between 112 and 212, then part of the foreground region pixels are misclassified as background. This is because the newly appearing pixels will be misclassified in the distribution which has a high variance, taking too long to update the variance value to its real value. Distributions with high weighting values tend to be classified as background.

4 Conclusions

The Gaussian mixture models are a type of density models which are composed of a number of

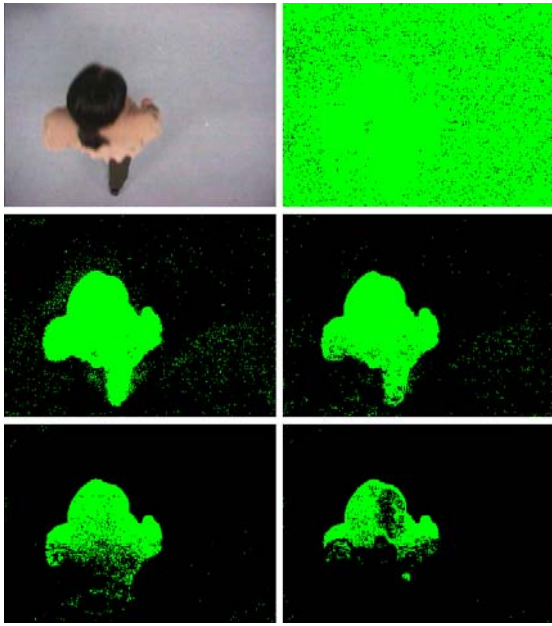


Figure 6: Top left: an original image of the sequence. Top right: result for standard deviation = 2. Middle left: standard deviation = 12. Middle right: standard deviation = 42. Bottom left: standard deviation = 112. Bottom right: standard deviation = 212.

components (functions). These functions can be used to model the colors of objects or backgrounds in a scene. This allows color-based object tracking and background segmentation. Adaptive Gaussian distributions are applicable for modelling changes, especially when related to fast moving objects such as vehicles on a highway.

The usage of Gaussian distributions has to be based on the application context. It can provide analysis results for long duration scenes (e.g., a surveillance system that monitors a car park or a campus day and night). It is also quite suitable for complex scenes or multi-colored objects. For outdoor scenes, different weather is taken into account. The Gaussian mixture model allows us to adapt to weather changes, such as from rain to snow, from cloudy to sunny, and so forth. Small movements in scenes like waving trees can also be handled. For simple indoor scenes or objects which appear to be monocolored, a small number of components in a Gaussian mixture model is suggested, say one or two components. For outdoor complex scenes, a larger number of components in a Gaussian mixture model is suggested, say starting with 3, but not extending 5 (very much). The maximum number is important if care has to be taken about computation time and system efficiency. In general, more components do have the potential for further improvement.

Of course, how to assign suitable values to parameters during an initialization period will also depend on specific applications. Values of parameters and other suit-

able initial values can be obtained during a pre-testing procedure. The higher the number of components of a mixture model, the better the results for a complex scene, but the computation time increases. Assigning a very small value to the learning rate will avoid that a slowly moving and large object melts into the background, but will affect the system's adaptation. One needs to balance out all these conditions according to different applications and environments.

References

- [1] Y. Bar-Shalom and X. R. Li. *Estimation and Tracking: Principles, Techniques, and Software*. Artech House, Boston, 1993.
- [2] C. J. Chang, W. F. Hu, J. W. Hsieh, and Y. S. Chen. Shadow elimination for effective moving object detection with Gaussian models. In Proc. *Int. Conf. Pattern Recognition*, 2: 540–543, 2002.
- [3] S. S. Cheung and C. Kamath: Robust techniques for background subtraction in urban traffic video. In Proc. *Electronic Imaging: Visual Comm. Image Proc.*, 881–892, 2004.
- [4] D. S. Lee: Effective Gaussian mixture learning for video background subtraction. *IEEE Trans. Pattern Analysis Machine Intelligence*, 27(5): 827–832, 2005.
- [5] S. J. McKenna, Y. Raja, and S. Gong. Object tracking using adaptive color mixture models. In Proc. *Asian Conf. Computer Vision*, 615–622, 1998.
- [6] Y. Raja, S. J. McKenna, and S. Gong. Tracking color objects using adaptive mixture models. In Proc. *Image Vision Computing*, 17: 225–231, 1999.
- [7] Y. Raja, S. J. McKenna, and S. Gong. Tracking and segmenting people in varying lighting conditions using color. In Proc. *IEEE Int. Conf. Automatic Face Gesture Recognition*, 228–233, 1998.
- [8] K. She, G. Bebis, H. Gu, and R. Miller: Vehicle tracking using on-line fusion of color and shape features. In Proc. *IEEE Int. Conf. Intelligent Transportation Systems*, 16: 731–736, 2004.
- [9] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In Proc. *Computer Vision and Pattern Recognition*, 2: 246–252, 1999.
- [10] Q. Zang and R. Klette. Evaluation of an adaptive composite Gaussian model in video surveillance. In Proc. *Image Vision Computing New Zealand*, 243–248, 2002.

Performance Evaluation of Accurate Ellipse Fitting

Kenichi Kanatani

Department of Computer Science, Okayama University, Okayama, 700-8530 Japan.

Email: kanatani@suri.it.okayama-u.ac.jp

Abstract

This paper studies numerical schemes for fitting an ellipse to points in an image. First, the problem is posed as maximum likelihood, and the relationship to the KCR lower bound is stated. Then, the algorithms of FNS, HEIV, renormalization, and Gauss-Newton iterations are described. Using simulated and real image data, their convergence properties are compared, and their dependence on the shape of the arc to which an ellipse is to be fitted is revealed.

Keywords: ellipse fitting, KCR lower bound, FNS, HEIV, renormalization

1 Introduction

Circular objects in the scene are generally projected onto ellipses on the image plane, and their 3-D positions can be computed from their images [6]. For this reason, fitting ellipses to a point sequence is one of the first steps of various vision applications. In this paper, we concentrate on numerical aspects, assuming that outliers have already been removed, e.g., by the procedure described in [10].

Various algebraic fitting methods were proposed in the past [1, 13, 15, 16], but Kanatani [8] pointed out that ellipse fitting can be regarded as statistical estimation and that maximum likelihood (ML) produces an optimal solution. Since then, many numerical schemes have been proposed, e.g., FNS [4], the HEIV [14], and Gauss-Newton iterations [11]. These methods attain a theoretical accuracy bound (KCR lower bound [3, 8]) up to high order terms in noise. Kanatani's renormalization [7, 8, 12] also computes a solution nearly equivalent to them [9].

All these methods are iterative, and the convergence properties are different from method to method. The purpose of this paper is to experimentally compare their convergence behavior.

2 Ellipse Fitting

An ellipse is represented by

$$Ax^2 + 2Bxy + Cy^2 + 2f_0(Dx + Ey) + Ff_0^2 = 0, \quad (1)$$

where f_0 is an arbitrary scaling constant¹. If we define

¹In our experiments, we set $f_0 = 600$. This is to make the coefficients have approximately the same magnitude for numerical stability. Theoretically, we can set $f_0 = 1$.

$$\mathbf{u} = (A, B, C, D, E, F)^\top, \quad (2)$$

$$\boldsymbol{\xi} = (x^2, 2xy, y^2, 2f_0x, 2f_0y, f_0^2)^\top, \quad (3)$$

Eq. (1) is written as

$$(\mathbf{u}, \boldsymbol{\xi}) = 0. \quad (4)$$

Throughout this paper, we denote the inner product of vectors \mathbf{a} and \mathbf{b} by (\mathbf{a}, \mathbf{b}) . Since the magnitude of the vector \mathbf{u} is indeterminate, we adopt normalization $\|\mathbf{u}\| = 1$.

Eq. (1) describes not necessarily an ellipse but also a parabola, a hyperbola, and their degeneracies (e.g., two lines) [6]. Even if the points (x_α, y_α) are sampled from an ellipse, the fitted equation may define a hyperbola or other curves in the presence of large noise, and a technique for preventing this has been proposed [13]. Here, however, we do not impose any constraints, assuming that noise is sufficiently small.

3 KCR Lower Bound

We write the data $\boldsymbol{\xi}_\alpha$ in the form $\boldsymbol{\xi}_\alpha = \bar{\boldsymbol{\xi}}_\alpha + \Delta\boldsymbol{\xi}_\alpha$, where $\bar{\boldsymbol{\xi}}_\alpha$ is the true value and $\Delta\boldsymbol{\xi}_\alpha$ the noise term. We define the covariance matrix of $\boldsymbol{\xi}_\alpha$ by

$$V[\boldsymbol{\xi}_\alpha] = E[\Delta\boldsymbol{\xi}_\alpha \Delta\boldsymbol{\xi}_\alpha^\top], \quad (5)$$

where $E[\cdot]$ denotes expectation over the noise distribution. If random noise of mean 0 and standard deviation σ is independently added to each coordinate of the points in the image, we can see from Eq. (3) that the covariance matrix $V[\boldsymbol{\xi}_\alpha]$ has the form $4\sigma^2 V_0[\boldsymbol{\xi}_\alpha]$ except for $O(\sigma^4)$, where $V_0[\boldsymbol{\xi}_\alpha]$ is

$$\begin{pmatrix} \bar{x}_\alpha^2 & \bar{x}_\alpha \bar{y}_\alpha & 0 & f_0 \bar{x}_\alpha & 0 & 0 \\ \bar{x}_\alpha \bar{y}_\alpha & \bar{x}_\alpha^2 + \bar{y}_\alpha^2 & \bar{x}_\alpha \bar{y}_\alpha & f_0 \bar{y}_\alpha & f_0 \bar{x}_\alpha & 0 \\ 0 & \bar{x}_\alpha \bar{y}_\alpha & \bar{y}_\alpha^2 & 0 & f_0 \bar{y}_\alpha & 0 \\ f_0 \bar{x}_\alpha & f_0 \bar{y}_\alpha & 0 & f_0^2 & 0 & 0 \\ 0 & f_0 \bar{x}_\alpha & f_0 \bar{y}_\alpha & 0 & f_0^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (6)$$

Here, $(\bar{x}_\alpha, \bar{y}_\alpha)$ is the true position of point (x_α, y_α) . In actual computations, $(\bar{x}_\alpha, \bar{y}_\alpha)$ is approximated² by the data position (x_α, y_α) .

We define the covariance matrix $V[\hat{\mathbf{u}}]$ of an estimate $\hat{\mathbf{u}}$ by

$$V[\hat{\mathbf{u}}] = E[(\mathbf{P}_\mathbf{u}\hat{\mathbf{u}})(\mathbf{P}_\mathbf{u}\hat{\mathbf{u}})^\top], \quad (7)$$

where $\mathbf{P}_\mathbf{u}$ is the projection matrix

$$\mathbf{P}_\mathbf{u} = \mathbf{I} - \mathbf{u}\mathbf{u}^\top, \quad (8)$$

which projects $\hat{\mathbf{u}}$ onto the hyperplane orthogonal to \mathbf{u} (\mathbf{I} denotes the unit matrix). Since the parameter vector \mathbf{u} is normalized to unit norm, its domain is the unit sphere \mathcal{S}^5 in \mathcal{R}^6 . We focus on the asymptotic limit of small noise and evaluate the error after projecting $\hat{\mathbf{u}}$ onto the tangent space to \mathcal{S}^5 at \mathbf{u} [8].

Kanatani [8, 9] proved that if ξ_α is regarded as an independent Gaussian random variable of mean $\bar{\xi}_\alpha$ and covariance matrix $V[\xi_\alpha]$, the following inequality holds for an arbitrary unbiased estimator $\hat{\mathbf{u}}$ of \mathbf{u} :

$$V[\hat{\mathbf{u}}] \succ 4\sigma^2 \left(\sum_{\alpha=1}^N \frac{\bar{\xi}_\alpha \bar{\xi}_\alpha^\top}{(\mathbf{u}, V_0[\xi_\alpha]\mathbf{u})} \right)_5^{-}. \quad (9)$$

Here, \succ means that the left-hand side minus the right is positive semidefinite, and $(\cdot)_r^{-}$ means pseudoinverse of rank r .

Chernov and Lesort [3] called the right-hand side of Eq. (9) the *KCR* (*Kanatani-Cramer-Rao*) *lower bound* and showed that it holds except for terms of $O(\sigma^4)$ even if $\hat{\mathbf{u}}$ is not unbiased; it is sufficient that $\hat{\mathbf{u}}$ is “consistent” in the sense that $\hat{\mathbf{u}} \rightarrow \mathbf{u}$ as $\sigma \rightarrow 0$.

4 Maximum Likelihood (ML)

Maximum likelihood (ML) under Gaussian noise assumption is to minimize the sum of squared Mahalanobis distances

$$J = \frac{1}{2} \sum_{\alpha=1}^N (\xi_\alpha - \bar{\xi}_\alpha, V_0[\xi_\alpha]_2^{-} (\xi_\alpha - \bar{\xi}_\alpha)), \quad (10)$$

subject to the constraints $(\mathbf{u}, \bar{\xi}_\alpha) = 0$, $\alpha = 1, \dots, N$. Eliminating the constraints by introducing Lagrange multipliers, we can write Eq. (10) as follows [8, 9]:

$$J = \frac{1}{2} \sum_{\alpha=1}^N \frac{(\mathbf{u}, \xi_\alpha)^2}{(\mathbf{u}, V_0[\xi_\alpha]\mathbf{u})}. \quad (11)$$

It can be shown that the covariance matrix $V[\hat{\mathbf{u}}]$ of the resulting estimator $\hat{\mathbf{u}}$ agrees with the KCR lower bound except for terms of $O(\sigma^4)$ [8, 9].

²We have confirmed that this does not cause any noticeable changes in the final results.

Eq. (11) is minimized by solving

$$\begin{aligned} \nabla_{\mathbf{u}} J &= \sum_{\alpha=1}^N \frac{(\mathbf{u}, \xi_\alpha) \xi_\alpha}{(\mathbf{u}, V_0[\xi_\alpha]\mathbf{u})} - \sum_{\alpha=1}^N \frac{(\mathbf{u}, \xi_\alpha)^2 V_0[\xi_\alpha]\mathbf{u}}{(\mathbf{u}, V_0[\xi_\alpha]\mathbf{u})^2} \\ &= (\mathbf{M} - \mathbf{L})\mathbf{u} = \mathbf{0}, \end{aligned} \quad (12)$$

where the 6×6 matrices \mathbf{M} and \mathbf{N} are defined by

$$\mathbf{M} = \sum_{\alpha=1}^N \frac{\xi_\alpha \xi_\alpha^\top}{(\mathbf{u}, V_0[\xi_\alpha]\mathbf{u})}, \quad (13)$$

$$\mathbf{L} = \sum_{\alpha=1}^N \frac{(\mathbf{u}, \xi_\alpha)^2 V_0[\xi_\alpha]}{(\mathbf{u}, V_0[\xi_\alpha]\mathbf{u})^2}. \quad (14)$$

FNS

The *FNS* (*fundamental numerical scheme*) of Chojnacki et al. [4] solves Eq. (12) by the following iterations:

1. Initialize \mathbf{u} .
2. Compute the matrices \mathbf{M} and \mathbf{L} in Eqs. (13) and (14).
3. Solve the eigenvalue problem

$$(\mathbf{M} - \mathbf{L})\mathbf{u}' = \lambda\mathbf{u}', \quad (15)$$

and compute the unit eigenvector \mathbf{u}' for the eigenvalue λ closest to 0.

4. If $\mathbf{u}' \approx \mathbf{u}$ except for sign, return \mathbf{u}' and stop. Else, let $\mathbf{u} \leftarrow \mathbf{u}'$ and go back to Step 2.

Later, Chojnacki et al. [5] pointed out that convergence performance improves if we choose in Step 3 not the eigenvalue closest to 0 but the smallest one. We call the above procedure the *original FNS* and the one using the smallest eigenvalue the *modified FNS*.

Whichever eigenvalue is chosen for λ , we have $\lambda = 0$ after convergence. In fact, convergence means

$$(\mathbf{M} - \mathbf{L})\mathbf{u} = \lambda\mathbf{u} \quad (16)$$

for some \mathbf{u} . Computing the inner product with \mathbf{u} on both sides, we have

$$(\mathbf{u}, \mathbf{M}\mathbf{u}) - (\mathbf{u}, \mathbf{L}\mathbf{u}) = \lambda. \quad (17)$$

On the other hand, Eqs. (13) and (14) imply that $(\mathbf{u}, \mathbf{M}\mathbf{u}) = (\mathbf{u}, \mathbf{L}\mathbf{u})$ identically, meaning $\lambda = 0$.

HEIV

Let

$$\xi_\alpha = \begin{pmatrix} z_\alpha \\ f_0^2 \end{pmatrix}, \quad \mathbf{u} = \begin{pmatrix} v \\ F \end{pmatrix}, \quad (18)$$

$$V_0[\xi_\alpha] = \begin{pmatrix} V_0[z_\alpha] & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{pmatrix}. \quad (19)$$

Define 5×5 matrices $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{L}}$ by

$$\tilde{\mathbf{M}} = \sum_{\alpha=1}^N \frac{\tilde{\mathbf{z}}_{\alpha} \tilde{\mathbf{z}}_{\alpha}^{\top}}{(\mathbf{v}, V_0[\mathbf{z}_{\alpha}]\mathbf{v})}, \quad (20)$$

$$\tilde{\mathbf{L}} = \sum_{\alpha=1}^N \frac{(\mathbf{v}, \tilde{\mathbf{z}}_{\alpha})^2 V_0[\mathbf{z}_{\alpha}]}{(\mathbf{v}, V_0[\mathbf{z}_{\alpha}]\mathbf{v})^2}, \quad (21)$$

where we put

$$\tilde{\mathbf{z}}_{\alpha} = \mathbf{z}_{\alpha} - \bar{\mathbf{z}}, \quad (22)$$

$$\bar{\mathbf{z}} = \sum_{\alpha=1}^N \frac{\mathbf{z}_{\alpha}}{(\mathbf{v}, V_0[\mathbf{z}_{\alpha}]\mathbf{v})} \bigg/ \sum_{\beta=1}^N \frac{1}{(\mathbf{v}, V_0[\mathbf{z}_{\beta}]\mathbf{v})}. \quad (23)$$

Then, Eq. (12) splits into the following two equations [5]:

$$\tilde{\mathbf{M}}\mathbf{v} = \tilde{\mathbf{L}}\mathbf{v}, \quad (\mathbf{v}, \bar{\mathbf{z}}) + f_0^2 F = 0. \quad (24)$$

If we determine a 5-D unit vector \mathbf{v} that satisfies the first equation, the value of F is determined from the second, and we obtain \mathbf{u} in the form

$$\mathbf{u} = N\left[\begin{pmatrix} \mathbf{v} \\ F \end{pmatrix}\right], \quad (25)$$

where $N[\cdot]$ denotes normalization to unit norm. The *HEIV* (*heteroscedastic errors-in-variables*) method of Leedan and Meer [14] computes the vector \mathbf{v} by the following iterations:

1. Initialize \mathbf{v} .
2. Compute the matrices $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{L}}$ in Eqs. (20) and (21).
3. Solve the generalized eigenvalue problem

$$\tilde{\mathbf{M}}\mathbf{v}' = \lambda \tilde{\mathbf{L}}\mathbf{v}', \quad (26)$$

and compute the unit eigenvector \mathbf{v}' for the eigenvalue λ closest to 1.

4. If $\mathbf{v}' \approx \mathbf{v}$ except for sign, return \mathbf{v}' and stop. Else, let $\mathbf{v} \leftarrow \mathbf{v}'$ and go back to Step 2.

However, Leedan and Meer [14] pointed out that choosing in Step 3 not the eigenvalue closest to 1 but the smallest one improves the convergence performance. We call the above procedure the *original HEIV* and the one using the smallest eigenvalue the *modified HEIV*.

Whichever eigenvalue is chosen for λ , we have $\lambda = 1$ after convergence. In fact, convergence means

$$\tilde{\mathbf{M}}\mathbf{v} = \lambda \tilde{\mathbf{L}}\mathbf{v} \quad (27)$$

for some \mathbf{v} . Computing the inner product with \mathbf{v} on both sides, we have

$$(\mathbf{v}, \tilde{\mathbf{M}}\mathbf{v}) = \lambda(\mathbf{v}, \tilde{\mathbf{L}}\mathbf{v}). \quad (28)$$

On the other hand, Eqs. (20) and (21) imply that $(\mathbf{v}, \tilde{\mathbf{M}}\mathbf{v}) = (\mathbf{v}, \tilde{\mathbf{L}}\mathbf{v})$ identically, meaning $\lambda = 1$.

Renormalization

The *renormalization* of Kanatani [8] is to approximate the matrix \mathbf{L} in Eq. (14) in the form

$$\mathbf{L} \approx c\mathbf{N}, \quad \mathbf{N} = \sum_{\alpha=1}^N \frac{V_0[\boldsymbol{\xi}_{\alpha}]}{(\mathbf{u}, V_0[\boldsymbol{\xi}_{\alpha}]\mathbf{u})}. \quad (29)$$

The constant c is determined so that $\mathbf{M} - c\mathbf{N}$ has eigenvalue 0. This is done by the following iterations [8]:

1. Initialize \mathbf{u} and let $c = 0$.
2. Compute the matrices \mathbf{M} and \mathbf{N} in Eqs. (13) and (29).
3. Solve the eigenvalue problem

$$(\mathbf{M} - c\mathbf{N})\mathbf{u}' = \lambda\mathbf{u}', \quad (30)$$

and compute the unit eigenvector \mathbf{u}' for the eigenvalue λ closest to 0.

4. If $\lambda \approx 0$, return \mathbf{u}' and stop. Else, let

$$c \leftarrow c + \frac{\lambda}{(\mathbf{u}', \mathbf{N}\mathbf{u}')}, \quad \mathbf{u} \leftarrow \mathbf{u}' \quad (31)$$

and go back to Step 2.

Gauss-Newton Iterations (GN)

Kanatani and Sugaya [11] proposed to minimize Eq. (11) directly by Gauss-Newton iterations. Differentiating Eq. (12) and introducing Gauss-Newton approximation (i.e., ignoring terms that contain $(\mathbf{u}, \boldsymbol{\xi}_{\alpha})$), we see that the Hessian is simply the matrix \mathbf{M} in Eq. (13). In order to enforce the normalization constraint $\|\mathbf{u}\| = 1$ in a differential form, we enforce \mathbf{M} to have eigenvalue 0 by the projection matrix $\mathbf{P}_{\mathbf{u}}$ of Eq. (8) and compute pseudoinverse. The procedure goes as follows:

1. Initialize \mathbf{u} .
2. Compute

$$\mathbf{u}' = N[\mathbf{u} - (\mathbf{P}_{\mathbf{u}}\mathbf{M}\mathbf{P}_{\mathbf{u}})^{-}_5(\mathbf{M} - \mathbf{L})\mathbf{u}]. \quad (32)$$

3. If $\mathbf{u}' \approx \mathbf{u}$, return \mathbf{u}' and stop. Else, let $\mathbf{u} \leftarrow \mathbf{u}'$ and go back to Step 2.

5 Initialization

For initialization of the iterations, we test the following three:

Random Choice

We generate six independent Gaussian random numbers of mean 0 and standard deviation 1 and normalize the vector consisting of them into unit norm.

Least Squares (LS)

Approximating the denominators in Eq. (11) by a constant, we minimize

$$J_{\text{LS}} = \frac{1}{2} \sum_{\alpha=1}^N (\mathbf{u}, \boldsymbol{\xi}_{\alpha})^2 = \frac{1}{2} (\mathbf{u}, \mathbf{M}_{\text{LS}} \mathbf{u}), \quad (33)$$

where we define

$$\mathbf{M}_{\text{LS}} = \sum_{\alpha=1}^N \boldsymbol{\xi}_{\alpha} \boldsymbol{\xi}_{\alpha}^{\top}. \quad (34)$$

Eq. (33) is minimized by the unit eigenvalue \mathbf{u} of \mathbf{M}_{LS} for the smallest eigenvalue.

Taubin's Method

Replacing the denominators in Eq. (11) by their average, we minimize the following function³ [16]:

$$J_{\text{TB}} = \frac{1}{2} \frac{\sum_{\alpha=1}^N (\mathbf{u}, \boldsymbol{\xi}_{\alpha})^2}{\sum_{\alpha=1}^N (\mathbf{u}, V_0[\boldsymbol{\xi}_{\alpha}] \mathbf{u})} = \frac{1}{2} \frac{(\mathbf{u}, \mathbf{M}_{\text{LS}} \mathbf{u})}{(\mathbf{u}, \mathbf{N}_{\text{TB}} \mathbf{u})}. \quad (35)$$

The matrix \mathbf{N}_{TB} has the form

$$\mathbf{N}_{\text{TB}} = \sum_{\alpha=1}^N V_0[\boldsymbol{\xi}_{\alpha}]. \quad (36)$$

Eq. (35) is minimized by solving the generalized eigenvalue problem

$$\mathbf{M}_{\text{LS}} \mathbf{u} = \lambda \mathbf{N}_{\text{TB}} \mathbf{u} \quad (37)$$

for the smallest eigenvalue. Since \mathbf{N}_{TB} is not positive definite, we decompose $\boldsymbol{\xi}_{\alpha}$, \mathbf{u} , and $V_0[\boldsymbol{\xi}_{\alpha}]$ in the form of Eqs. (19) and define 8×8 matrices $\tilde{\mathbf{M}}_{\text{LS}}$ and $\tilde{\mathbf{N}}_{\text{TB}}$ by

$$\tilde{\mathbf{M}}_{\text{LS}} = \sum_{\alpha=1}^N \tilde{z}_{\alpha} \tilde{z}_{\alpha}^{\top}, \quad \tilde{\mathbf{N}}_{\text{TB}} = \sum_{\alpha=1}^N V_0[\mathbf{z}_{\alpha}], \quad (38)$$

where

$$\tilde{z}_{\alpha} = \mathbf{z}_{\alpha} - \bar{\mathbf{z}}, \quad \bar{\mathbf{z}} = \frac{1}{N} \sum_{\alpha=1}^N \mathbf{z}_{\alpha}. \quad (39)$$

Then, Eq. (37) splits into two equations

$$\tilde{\mathbf{M}}_{\text{LS}} \mathbf{v} = \lambda \tilde{\mathbf{N}}_{\text{TB}} \mathbf{v}, \quad (\mathbf{v}, \bar{\mathbf{z}}) + f_0^2 F_{33} = 0. \quad (40)$$

We compute the unit eigenvector \mathbf{v} of the first equation for the smallest eigenvalue λ . The second equation gives F_{33} , and \mathbf{u} is given by Eq. (25).

³Taubin [16] did not take the covariance matrix into account. This is a modification of his method.

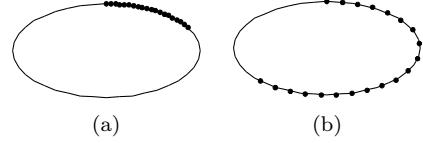


Figure 1: 20 points on elliptic arcs. (a) Short arch. (b) Long arc

6 Numerical Examples

Fig. 1 shows two examples of 20 equidistant points $(\bar{x}_{\alpha}, \bar{y}_{\alpha})$ on an ellipse. We added Gaussian noise of mean 0 and standard deviation σ to the x and y coordinates of each point independently and fitted an ellipse by FNS, HEIV, renormalization, and GN. For each σ , we plotted the average number of iterations over 1000 independent trials. We stopped when the new value \mathbf{u}' differs from the previous value⁴ \mathbf{u} by $\|\mathbf{u}' - \mathbf{u}\| < 10^{-6}$.

Doing numerical experiments, we have found that the convergence performance significantly differs depending on whether we use points on a short elliptic arc or on a long elliptic arc.

Fitting to a Short Arc

Figure 2 plots the number of iterations for the short arc in Fig. 1(a). When the iterations did not converge after 100 iterations, we stopped and set the iteration count to 100. We can see that the modified FNS/HEIV always converge faster than the original FNS/HEIV. This is most apparent for random initialization, for which the original FNS/HEIV did not converge for 16% and 49%, respectively, of the trials.

This can be explained as follows. If the computed \mathbf{u}' is close to the true value \mathbf{u} , the matrix \mathbf{L} in Eq. (14) and the matrix $\tilde{\mathbf{L}}$ in Eq. (21) are both close to \mathbf{O} . Initially, however, they may be very different from \mathbf{O} . Eqs. (15) and (26) are written, respectively, as

$$(\mathbf{M} - \mathbf{L} - \lambda \mathbf{I}) \mathbf{u}' = \mathbf{0}, \quad (\tilde{\mathbf{M}} - \lambda \tilde{\mathbf{L}}) \mathbf{v}' = \mathbf{0}. \quad (41)$$

The matrices \mathbf{L} and $\tilde{\mathbf{L}}$ are both positive definite. In order that their effects be canceled, we need to choose λ to be negative in the first equation and smaller than 1 in the second.

As predicted from this explanation, the difference between the original FNS/HEIV and the modified FNS/HEIV shrinks as we use better initial values, as seen from Fig. 2.

Another finding is that although FNS, HEIV and GN converges faster as we use better initial values, the behavior of renormalization is almost unchanged. This is because we start solving Eq. (30)

⁴Since \mathbf{u} and $-\mathbf{u}$ represent the same ellipse, we computed the smaller of the two values $\|\mathbf{u}' \pm \mathbf{u}\|$.

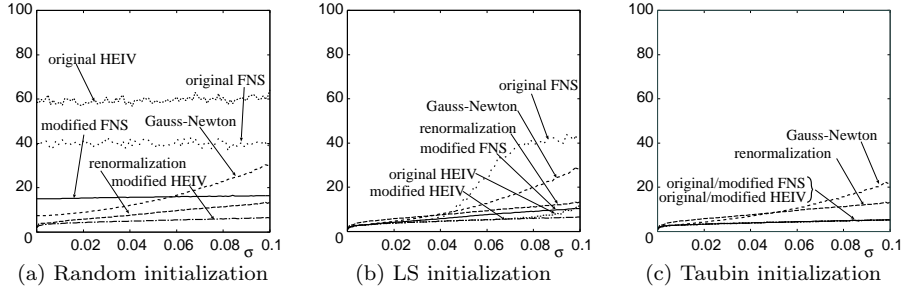


Figure 2: Average number of iterations for ellipse fitting to the points in Fig. 1(a) vs. noise level.

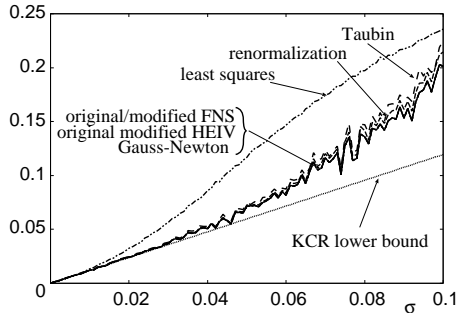


Figure 3: Root-mean-squares error of ellipse fitting to the points in Fig. 1(a) vs. noise level.

with $c = 0$, canceling the effect of \mathbf{N} whatever it is, and the resulting \mathbf{u}' is close to the LS solution.

In contrast, FNS and HEIV may produce a solution very different from the true value when initially the matrices \mathbf{L} and $\tilde{\mathbf{L}}$ are very different from \mathbf{O} . Naturally, GN converges faster if started from better initial values.

Overall, the most efficient method is the modified HEIV for whichever initialization. However, there is no difference between (original or modified) FNS/HEIV if initialized by Taubin’s method.

Fig. 3 plots for each σ the root-mean-squares of $\|\mathbf{P}_u \hat{\mathbf{u}}\|$ over 1000 independent trials. We compared LS, Taubin’s method, and the four iterative methods starting from the Taubin solution. We confirmed that for each method the final solution does not depend on the initial value as long as the iterations converge. The dotted line indicates the KCR lower bound implied by Eq. (9).

From Fig. 3, we can see that Taubin’s method is considerably better⁵ than LS. The four iterative methods indeed improve the Taubin solution, but the improvement is rather small. All the solutions nearly agree with the KCR lower bound when noise is small; as noise increases, they gradually deviate from it. Since FNS, HEIV, and GN minimize the same function, the resulting solution is virtually the same. The accuracy of renormalization is also very close to them.

⁵The mechanism of the superiority of Taubin’s method over LS is analyzed in detail in [9].

Fitting to a Long Arc

Fig. 4 shows the number of iterations for the long arc in Fig. 1(b). In this case, all methods converged within 10 iterations when initialized by LS or Taubin’s method, so the vertical axis is restricted over that range.

The most unexpected, as compared with Fig. 2, is the fact that *the modified FNS is worse than the original FNS*. For random initialization, the modified FNS did not converge after 100 iterations for *all* 1000 trials, while the original FNS failed to converge only for 24% of the trials.

This is related to the singularity of ellipse fitting [2]: Some of the terms on the right-hand side of Eq. (11) diverge to $\pm\infty$. This happens when a data point exists near the center of the current candidate fit, which is more likely to occur when the data points are distributed over a long arc.

As we can see from Fig. 4, renormalization is the most stable for whichever initialization. As we noted earlier, this is because the iterations start from $c = 0$; Eq. (30) yields a value \mathbf{u}' close to the LS solution, which is already fairly accurate for a long arc. GN is also stable, because the solution continuously changes in the course of the iterations, while FNS and HEIV may compute oscillating eigenvectors.

Figure 5 compares the accuracy of all the methods in the same way as Fig. 3. As expected, the LS solution, which is usually prone to statistical bias, is as accurate as Taubin’s method, because bias is less likely to arise for a long arc. Also, the improvement by the (original or modified) FNS/HEIV, renormalization, and GN is very small. All yields practically the same solution very close to the KCR lower bound.

7 Conclusions

We have studied the convergence behavior of typical iterative numerical schemes for maximal likelihood (ML) of ellipse fitting. After posing the problem in relation to the KCR lower bound, we

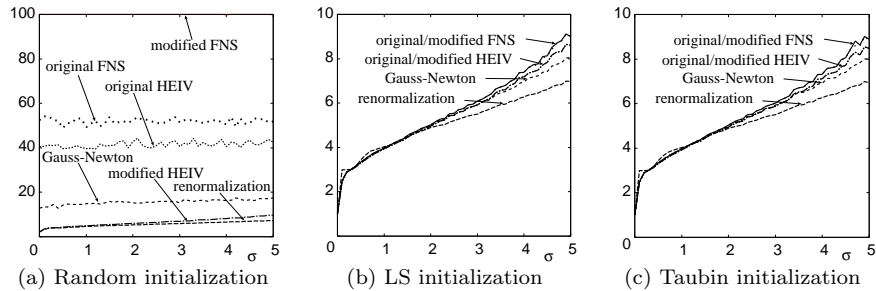


Figure 4: Average number of iterations for ellipse fitting to the points in Fig. 1(b) vs. noise level.

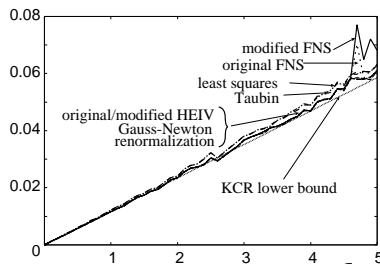


Figure 5: Root-mean-squares error of ellipse fitting to the points in Fig. 1(b) vs. noise level.

described the algorithms of FNS, HEIV, renormalization, and Gauss-Newton iterations (GN). Using simulated image data, we compared their convergence performance.

For a short arc, the modified FNS/HEIV have better convergence properties than the original FNS/HEIV. The convergence of renormalization is little affected by the choice of the initial value. Overall, the modified HEIV is the most efficient.

For a long arc, however, the modified FNS is worse than the original FNS if randomly initialized, and the renormalization is the most efficient. If the iterations converge, however, the fitting accuracy is far higher than for a short arc whichever method is used.

Acknowledgments. The authors thanks Nikolai Chernov of the University of Alabama, U.S.A., and Wojciech Chojnacki of the University of Adelaide, Australia, for helpful discussions. He also thank Yasuyuki Sugaya of Toyohashi University of Technology, Japan, and Junpei Yamada of Okayama University, Japan, for participating in numerical experiments. This work was supported in part by the Ministry of Education, Culture, Sports, Science and Technology, Japan, under a Grant in Aid for Scientific Research C (No. 17500112).

References

- [1] F.J. Bookstein, Fitting conic sections to scattered data, *Comput. Graphics Image Process.*, **9** (1979), 56–71.
- [2] N. Chernov, On the convergence of numerical schemes in computer vision, *J. Math. Imaging. Vision*, **25** (2007), to appear.
- [3] N. Chernov and C. Lesort, Statistical efficiency of curve fitting algorithms, *Comput. Stat. Data Anal.*, **47-4** (2004-11), 713–728.
- [4] W. Chojnacki, M.J. Brooks, A. van den Hengel and D. Gawley, On the fitting of surfaces to data with covariances, *IEEE Trans. Patt. Anal. Mach. Intell.*, **22-11** (2000-11), 1294–1303.
- [5] W. Chojnacki, M. J. Brooks, A. van den Hengel and D. Gawley, FNS, CFNS and HEIV: A unifying approach, *J. Math. Imaging Vision*, **23-2** (2005-9), 175–183.
- [6] K. Kanatani, *Geometric Computation for Machine Vision*, Oxford University Press, Oxford, U.K., 1993.
- [7] K. Kanatani, Statistical bias of conic fitting and renormalization, *IEEE Tran. Patt. Anal. Mach. Intell.*, **16-3** (1994-3), 320–326.
- [8] K. Kanatani, *Statistical Optimization for Geometric Computation: Theory and Practice*, Elsevier Science, Amsterdam, The Netherlands, 1996; reprinted, Dover, New York, 2005.
- [9] K. Kanatani, Hyperaccuracy for geometric fitting, *4th Int. Workshop Total Least Squares and Errors-in-Variables Modelling*, Leuven, Belgium, August 2006.
- [10] K. Kanatani and N. Ohta, Automatic detection of circular objects by ellipse growing, *Int. J. Image Graphics*, **4-1** (2004-1), 35–50.
- [11] K. Kanatani and Y. Sugaya, High accuracy fundamental matrix computation and its performance evaluation, *Proc. 17th British Machine Vision Conf.*, Sept. 2006, Edinburgh, U.K., Vol. 1, pp. 217–226.
- [12] Y. Kanazawa and K. Kanatani, Optimal conic fitting and reliability evaluation, *IEICE Trans. Inf. & Sys.*, **E79-D-9** (1996-9), 1323–1328.
- [13] A. Fitzgibbon, M. Pilu and R.B. Fisher, Direct least square fitting of ellipses, *IEEE Trans. Patt. Anal. Mach. Intell.*, **21-5** (1999-3), 476–480.
- [14] Y. Leedan and P. Meer, Heteroscedastic regression in computer vision: Problems with bilinear constraint, *Int. J. Comput. Vision.*, **37-2** (2000-6), 127–150.
- [15] P.D. Sampson, Fitting conic sections to “very scattered” data: An iterative refinement of the Bookstein algorithm, *Comput. Graphics Image Process.*, **18** (1982), 97–108.
- [16] G. Taubin, Estimation of planar curves, surfaces, and non-planar space curves defined by implicit equations with applications to edge and range image segmentation, *IEEE Trans. Patt. Anal. Mach. Intell.*, **13-11** (1991-11), 1115–1138.

Rectifying Images for Stereo Vision

Y. Lin, A. Woodward, D. An, J. Morris, P. Delmas, and G. Gimel'farb

Department of Computer Science, Tamaki Campus, University of Auckland.

Email: awoo016@ec.auckland.ac.nz

Abstract

Two image rectification methods for stereo vision are presented – the first using a calibration result and the second a new approach relying on point correspondences. Both methods use a linear transformation and retain camera optical centres. These methods are proposed for rectifying weakly aligned or convergent camera setups, as found in many laboratory settings. Rectified results can be directly used in disparity map generation or 3D reconstruction. Experimental results show that both methods are suitable for rectifying images for input into stereo vision algorithms.

Keywords: Image rectification, stereo vision, epipolar geometry, fundamental matrix, essential matrix, camera calibration

1 Introduction

This work presents two image rectification methods – the first using a calibration result and the second a new approach relying on point correspondences. Both methods use a linear transformation and retain the optical centres, so rectified results can be directly used in disparity map generation or 3D reconstruction. The aim is to rectify weakly aligned or convergent cameras as opposed to arbitrary camera configurations.

Both methods assume camera intrinsic parameters are known and incorporate lens distortion correction by using calibration results or an independent distortion measurement. Image resampling errors are reduced by performing distortion correction and rectification concurrently.

Many stereo vision algorithms assume cameras generate an image pair which satisfies a standard stereo geometry. Namely, for a given point in one image, its corresponding point lies on the same scanline in the second image. This property greatly speeds up the stereo correspondence process as it reduces the search space to one dimension. Image rectification transforms and resamples an image pair so they have this desired rectilinear property.

Section 2 presents related work. Sections 3 and 4 describe the two proposed rectification methods, followed by experimental results and conclusion in Sections 5 and 6.

2 Related work

The careful mechanical alignment of two identical cameras to standard stereo geometry is difficult

and time consuming. Aligning two high-resolution cameras requires painstaking care, even when they are attached to precisely adjustable bases and mounted on a precise translation rail. There are also cases where rectification is necessary, e.g. images taken by weakly aligned cameras, structure from motion, aerial photography, and other applications where precise alignment is not possible.

Various techniques exist for image rectification. Some require camera calibration, such as the algorithm of Ayache et al. [1], which uses knowledge of camera projection matrices. In contrast, some rely solely on point correspondences, e.g. the methods of Hartley [6], and Loop and Zhang [3]. Methods relying on point correspondences do not retain the baseline length and scenes can only be reconstructed up to 3D projectivity, where angles and relative lengths are not preserved [7]. The method of Oram [4], which aims to rectify arbitrary epipolar geometry, uses nonlinear rectification which could introduce image distortion.

3 Rectification using camera calibration results

In standard stereo geometry, two identical cameras should be aligned so that both image planes are coplanar and their x-axes are parallel to the baseline.

To satisfy this requirement, an intuitive method is to rotate both cameras around their optical centres to a common orientation. Using calibration results, one can compute the baseline and a new common

orientation from the poses, or relative poses, of the two cameras.

This is similar to Ayache et al.'s method [1]. However, their method used the 3×4 camera projection matrix. We use the calibrated extrinsic and intrinsic camera parameters, which simplifies calculations and decouples lens distortion from the projection matrix.

3.1 Obtaining a common orientation

Let the calibrated rotation matrices and translation vectors of two cameras be \mathbf{R}_1 , \mathbf{R}_2 , \mathbf{t}_1 , and \mathbf{t}_2 respectively, defined in a world coordinate system. Their optical centres, \mathbf{c}_1 and \mathbf{c}_2 , are then defined as

$$\begin{aligned} \mathbf{c}_1 &= -\mathbf{t}_1 \mathbf{R}_1^T \\ \mathbf{c}_2 &= -\mathbf{t}_2 \mathbf{R}_2^T \end{aligned} \quad (1)$$

The baseline vector \mathbf{b} is

$$\mathbf{b} = \mathbf{c}_2 - \mathbf{c}_1 \quad (2)$$

The optical axis (z-axis) of camera one, \mathbf{z}_1 , is the world coordinate representation of the axis $(0, 0, 1)^T$ in the camera coordinate frame:

$$\mathbf{z}_1 = \mathbf{R}_1^T \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad (3)$$

which is exactly the third row of \mathbf{R}_1 . In the same manner, the x-axis and y-axis of camera one are respectively the first and second row of \mathbf{R}_1 .

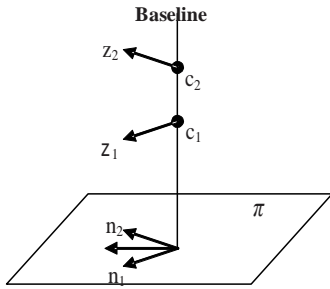


Figure 1: Finding a new common orientation

Let the plane π be the plane perpendicular to the baseline. The optical axes, \mathbf{z}_1 and \mathbf{z}_2 are projected onto π as shown in Figure 1. The directions of \mathbf{n}_1 , \mathbf{n}_2 are given by

$$\begin{aligned} \mathbf{n}_1 &= (\mathbf{b} \times \mathbf{z}_1) \times \mathbf{b} \\ \mathbf{n}_2 &= (\mathbf{b} \times \mathbf{z}_2) \times \mathbf{b} \end{aligned} \quad (4)$$

It is clear that both \mathbf{n}_1 and \mathbf{n}_2 will be perpendicular to the baseline. \mathbf{n}_1 or \mathbf{n}_2 may be used

as the new common optical axis, \mathbf{z}_{new} . But for a better range of common view, the half vector between \mathbf{n}_1 and \mathbf{n}_2 is chosen. After \mathbf{n}_1 , \mathbf{n}_2 have been normalised, \mathbf{z}_{new} is simply their average (and must also be normalised):

$$\mathbf{z}_{new} = \frac{(\mathbf{n}_1 + \mathbf{n}_2)}{|\mathbf{n}_1 + \mathbf{n}_2|} \quad (5)$$

Since both camera's x-axes should run along the baseline, the x-axis of the new orientation \mathbf{x}_{new} is the same as the baseline \mathbf{b} . Assuming camera one is on the left side of camera two, then:

$$\mathbf{x}_{new} = \frac{\mathbf{b}}{|\mathbf{b}|} \quad (6)$$

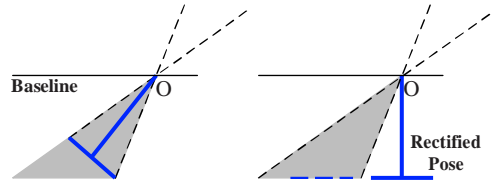


Figure 2: Shifting into the view frustum

The new y-axis, \mathbf{y}_{new} , is the cross product of \mathbf{x}_{new} and \mathbf{z}_{new}

$$\mathbf{y}_{new} = \mathbf{x}_{new} \times \mathbf{z}_{new} \quad (7)$$

After normalising the \mathbf{x}_{new} , \mathbf{y}_{new} and \mathbf{z}_{new} vectors, the matrix representation of the new orientation \mathbf{R}_{new} is

$$\mathbf{R}_{new} = (\mathbf{x}_{new} \ \mathbf{y}_{new} \ \mathbf{z}_{new})^T \quad (8)$$

The rotation matrices \mathbf{M}_1 , \mathbf{M}_2 that rotate the cameras to their new orientations are given by

$$\begin{aligned} \mathbf{M}_1 &= \mathbf{R}_{new} \mathbf{R}_1^T \\ \mathbf{M}_2 &= \mathbf{R}_{new} \mathbf{R}_2^T \end{aligned} \quad (9)$$

3.2 Rotation of images

According to Hartley [8], when rotating two cameras around their optical centres, their images will be transformed by the rotation homographies \mathbf{H}_1 , \mathbf{H}_2 :

$$\begin{aligned} \mathbf{H}_1 &= \mathbf{K}_1 \mathbf{M}_1 \mathbf{K}_1^{-1} \\ \mathbf{H}_2 &= \mathbf{K}_2 \mathbf{M}_2 \mathbf{K}_2^{-1} \end{aligned} \quad (10)$$

where \mathbf{K}_1 , \mathbf{K}_2 are the intrinsic matrices of cameras one and two.

The geometric meaning of Equation 10 is that \mathbf{K}^{-1} transforms a point in the 2D image to a 3D normalised camera coordinate, that is then rotated by \mathbf{M} , followed by \mathbf{K} projecting the point back onto the image plane.

In some cases, especially in a convergent camera setup, the rotation angle around the y-axis may be large, causing the rotated image plane to be out of the original view frustum, as shown in Figure 2. To obtain a larger view, one may shift the centre of the image plane into the frustum. This shift will not violate the pinhole camera model, but it must be added back to any disparities calculated from the rectified images.

3.3 Concurrent correction of lens distortion

Once the transformation matrix \mathbf{H} has been calculated, an image point \mathbf{p} in the original image is transformed to \mathbf{p}' in the new image by

$$\mathbf{p}' = \mathbf{H}\mathbf{p} \quad (11)$$

Image resampling uses bilinear interpolation and backwards mapping. For the backwards mapping, the value at pixel \mathbf{p}' in the new image is obtained from point \mathbf{p} in the original image (see Figure 3). \mathbf{p} can be calculated by

$$\mathbf{p} = \mathbf{H}^{-1}\mathbf{p}' \quad (12)$$

Distortion correction is integrated at this stage – the pixel value is read from the distorted coordinate of \mathbf{p} , instead of the value at \mathbf{p} .

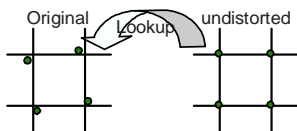


Figure 3: Backward mapping for image resampling

4 Rectification using point correspondences

Uncalibrated image rectification commonly uses the fundamental matrix. Usually a pair of transformations are found which are compatible with the fundamental matrix and send the epipoles to infinity [6]. However, these transformations make no effort to retain the baseline and there will be an undesirable projective reconstruction ambiguity in the result.

Assuming the intrinsic camera parameters are known or can be determined, it is possible to back project image points into a normalised camera

coordinate frame. A linear rectification is then to find rotations around two optical centres that make the images conform to the desired standard stereo geometry.

Hence, the problem becomes finding the Euler rotation angles, R_x, R_y, R_z , of camera one and two that minimise the total vertical displacement error of corresponding points.

4.1 Finding a rectification homography

Estimation proceeds by first preparing a set of rotation angles, $\{R_{1x}, R_{1y}, R_{1z}, R_{2x}, R_{2y}, R_{2z}\}$, and initially setting them to zero (or values to be discussed in Section 4.2).

Two rotation matrices, \mathbf{R}_1 and \mathbf{R}_2 , are constructed from the set of rotation angles. Then all corresponding points are transformed using one of the rotation matrices, \mathbf{R} , and the intrinsic matrix \mathbf{K} :

$$\mathbf{p}' = \mathbf{K}\mathbf{R}\mathbf{K}^{-1}\mathbf{p} \quad (13)$$

Homogenous coordinates are then transformed into Euclidean coordinates:

$$\begin{aligned} p'_x &= p'_x/p'_z \\ p'_y &= p'_y/p'_z \end{aligned} \quad (14)$$

The error being minimised is the vertical displacement of corresponding points:

$$\mathbf{e} = p'_{1y} - p'_{2y} \quad (15)$$

The pair of rotation matrices that minimise the sum of squared errors for i points is

$$(\mathbf{R}_1, \mathbf{R}_2) = \underset{(\mathbf{R}_1, \mathbf{R}_2)}{\operatorname{argmin}} \sum_i e_i^2 \quad (16)$$

Minimisation can be done by steepest descent or other iterative estimation methods. Here, the Levenberg-Marquardt method (LM) was used. After minimisation, a rotation around the x-axis can be applied to both images in order to have a larger common field of view.

Experiments showed that if the cameras had been weakly aligned, setting all initial values to zero usually lead to satisfactory results. However, with images taken by convergent cameras or hand held cameras, a good initial guess of the rotation angles is required to avoid the solution being trapped in a local minima. With known intrinsic parameters and an accurate fundamental matrix, the essential matrix \mathbf{E} and subsequently the relative orientation of two cameras can be estimated.

4.2 Estimating the relative pose from the essential matrix

The essential matrix, \mathbf{E} , is a specialisation of the fundamental matrix and encodes the baseline (up to a scale factor) and relative orientation of two cameras [6]. It is given as

$$\mathbf{E} = [\mathbf{b}]_{\times} \mathbf{R} \quad (17)$$

where $[\mathbf{b}]_{\times}$ is the skew-symmetric matrix of the baseline vector, \mathbf{b} , and \mathbf{R} is the relative orientation.

The essential matrix can be calculated from the fundamental matrix, \mathbf{F} , together with the intrinsic parameter matrices, \mathbf{K}_1 and \mathbf{K}_2 , of two cameras:

$$\mathbf{E} = \mathbf{K}_2^T \mathbf{F} \mathbf{K}_1 \quad (18)$$

To decompose the essential matrix to a skew-symmetric matrix and an orthonormal rotation matrix, Horn [2] described an approach in which the baseline, \mathbf{b} , is calculated as

$$\mathbf{b}\mathbf{b}^T = \frac{1}{2} \text{Trace}(\mathbf{E}\mathbf{E}^T) \mathbf{I} - \mathbf{E}\mathbf{E}^T \quad (19)$$

where \mathbf{I} is the 3×3 identity matrix, and the relative rotation, \mathbf{R} , as

$$(\mathbf{b} \cdot \mathbf{b})\mathbf{R} = \text{Cofactors}(\mathbf{E})^T - [\mathbf{b}]_x \mathbf{E} \quad (20)$$

where $\text{Cofactors}(\mathbf{E})$ is the matrix of cofactors of \mathbf{E} . Once the direction of the baseline and the relative orientation are found, rotations to a common orientation can be calculated as described in Section 4.1. The rotation angles calculated from the fundamental matrix can serve as good initial values for the Levenberg-Marquardt minimisation.

5 Experimental Results

5.1 Rectification from calibration

In this section, two camera models were tested. The models and their setup are listed in Table 1.

Model	Image Size	Focus	Speed	f stop	Format	Mounting
Canon A80	1600 x 1200	19 mm	1/30s	5.6	JPG	Hand held
Canon EOS 10D	3056 x 2048	51 mm	1/125s	4.0	RAW	On Rail

Table 1: Cameras used in the experiment

Figure 4 shows a pair of images taken by the same Canon A80 camera. The camera was hand-held and images were taken with a convergent angle of about 12 degrees. They show a non-coplanar calibration cube with 63 patches for feature extraction.

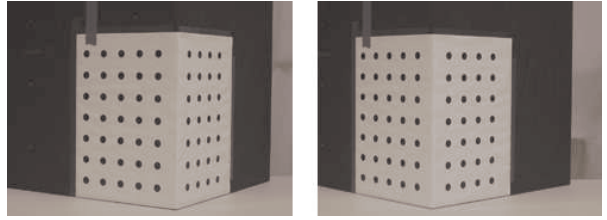


Figure 4: Original images of calibration cube, taken by the Canon A80

The Tsai camera calibration algorithm [9] was used and rectification matrices were calculated from the calibration results, as discussed in Section 3. The image pair was resampled and all feature points remeasured from the resultant images. Statistics of vertical displacement errors for 63 pairs of corresponding points are shown in Table 2.

Canon A80 1600 x 1200	Average(pix)	Stdev	Max(pix)
Original image pair	-2.56	6.25	15.01
Rectified	0.03	0.25	0.59
Rectified & distortion corrected	0.01	0.10	0.30

Table 2: Vertical displacements of corresponding pairs: Canon A80

Canon EOS 10D 3056 x 2048	Average(pix)	Stdev	Max(pix)
Original image pair	-2.70	3.58	8.03
Rectified	0.34	0.34	0.99
Rectified & distortion corrected	0.01	0.12	0.36

Table 3: Vertical displacements of corresponding pairs: Canon EOS 10D

The second pair of images was taken by two Canon EOS 10D cameras mounted on a translational rail. The baseline length was about 32 cm. The same calibration routine and rectification method was used and Table 3 shows the results.

After rectification, the y-axis displacement error decreased to less than one pixel. Distortion correction further decreased the error in both image sets. The gain is more significant in the higher resolution image set, the maximum error is down to about 0.4 pixels and the standard deviation to about 0.1 pixels. Noting the high resolution of the images, these results are reasonable for use in stereo correspondence searching.

5.2 Rectification from point correspondences

The two sets of images from the first experiment (Section 5.1) were again used. However, calibration results were discarded and the intrinsic matrices were constructed from camera specifications and effective focal length (omitting the scale factor):

$$\mathbf{K} = \begin{pmatrix} f/w & 0 & C_x \\ 0 & f/h & C_y \\ 0 & 0 & 1 \end{pmatrix} \quad (21)$$

where w and h are the width and height of a sensor grid in the camera specification, f the focal length, and (C_x, C_y) the image centre.

Canon A80 1600 x 1200	Average(pixel)	Stdev	Max(pixel)
Original image pair	-2.56	6.52	15.01
Rectified	0.00	0.07	0.15
Rectified & distortion corrected	0.00	0.07	0.15

Table 4: Rectification result (vertical displacements) of LM minimisation: Canon A80

Canon EOS 10D 3056 x 2048	Average(pixel)	Stdev	Max(pixel)
Original image pair	-2.70	3.58	8.30
Rectified	0.00	0.18	0.46
Rectified & distortion corrected	0.00	0.15	0.37

Table 5: Rectification result (vertical displacements) of LM minimisation: Canon EOS 10D

All initial values were set to zero before LM minimisation since the needed rectifying rotation angles were small. Tables 4 and 5 show the y-axis displacement errors before and after rectification.

Although results were comparative to the first method, it is notable that the rotation angles obtained by the minimisation did not guarantee two cameras rotated to a common orientation.

The fundamental matrix decomposition approach faces further challenges – the result of the decomposition greatly depends on the accuracy of the fundamental matrix, something not always easily achievable.

6 Conclusion

Two image rectification methods were presented and tested for rectifying images taken by convergent or weakly aligned cameras. The first method used calibration results and rotated the images around their optical centres to a common orientation. The second method is a new approach based on point correspondences with an assumption that intrinsic camera parameters are obtainable. Both methods retain the baseline and the rectifying transformations are linear. This allows the rectified results to be used directly for 3D reconstruction without projectivity adjustment¹.

Lens distortion correction was incorporated into the rectification process. Experiments showed both methods to be efficient in rectifying

¹Parallelism remains but angles and lengths do not in projectivity reconstructions.

images taken by weakly aligned and convergent camera setups. After rectification and distortion correction, the maximum y-axis displacement error was less than 0.4 pixels in the resampled images. This can be considered reasonable for scanline searching in the stereo correspondence process.

The work documented here has been successfully used in Woodward et al. [10].

References

- [1] N. Ayache and C. Hansen, “Rectification of Images for Binocular and Trinocular Stereovision”, in *Proc. 9th International Conference on Pattern Recognition, Rome, 1988*.
- [2] B.K.P. Horn, “Recovering baseline and orientation from essential matrix”, *Available Jan. 2006 at <http://www.ai.mit.edu/people/bkph/publications.html>, 1990*.
- [3] C. Loop and Z. Zhang, “Computing rectifying homographies for stereo vision”, in *Proc. of the 1999 Conference on Computer Vision and Pattern Recognition, 1999*.
- [4] D. Oram, “Rectification for any Epipolar Geometry”, in *12th British Machine Vision Conference (BMVC 2001), September 2001*
- [5] Z. Hang, “On the Epipolar Geometry Between Two Images With Lens Distortion”, in *Proc. Int’l Conf. Pattern Recognition (ICPR), Vol. I, pp. 407-411, Vienna, Aug. 1996*.
- [6] R. Hartley, “Theory and Practice of Projective Rectification”, in *International Journal of Computer Vision, Vol. 35, No. 2, pp. 115-127, November 1999*.
- [7] R. Hartley and A. Zisserman. “Multiple View Geometry in Computer Vision second edition”, Cambridge Press 2003.
- [8] R. Hartley, “Self-calibration of stationary cameras”, *Int. J. Comput. Vision 22, no. 1, 5-23, 1997*.
- [9] R. Y. Tsai, “A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses”, in *IEEE Journal of Robotics and Automation RA-3(4), 1987*.
- [10] A. Woodward, D. An, Y. Lin, P. Delmas, G. Gimel’farb, and J. Morris, “An Evaluation of Three Popular Computer Vision Approaches for 3-D Face Synthesis”, in *Joint International Workshops on Structural, Syntactic and Statistical Pattern Recognition Hong Kong, China, August 17-19, 2006*.

An Image Data Hiding Scheme being Perfectly Imperceptible to Histogram Attacks

Hung-Min Sun¹, Yao-Hsin Chen², King-Hang Wang³

Department of Computer Science
National Tsing Hua University
Hsinchu, Taiwan

Email: ¹humsun@cs.nthu.edu.tw, {²saint_chen, ³khwang0}@is.cs.nthu.edu.tw

Abstract

Data hiding schemes using least-significant-bit (LSB) substitution can be steganalyzed with histogram of the pixels. This is because LSB substitution causes a *Pairs-of-Value* (PoV) effect to the histogram. Some works have been proposed before to improve the situation. However, all of those cannot avoid changing the histogram, in which, can still be suffered from other statistical attack in histogram. In this paper, we propose a novel approach in image data hiding without affecting the histogram. Experimental results show our method has fairly good data hiding capacity and low noise level.

Keywords: Image data hiding, steganalysis, histogram analysis, LSB substitution, Pairs-of-Values.

1 Introduction

The aim of image data hiding is to hide information imperceptibly into a host image, so that the presence of hidden data cannot be identified. Generally, a good steganography technique should have good visual and statistical imperceptibility. The algorithm to detect whether an image is loaded with secret data is called steganalysis. A successful steganalysis algorithm should have a low false-positive rate and false-negative rate.

LSB substitution is known as the simplest scheme in steganography. This algorithm replaces the LSB of the host image with the secret data stream. To avoid data extraction by adversary, the data stream should be encrypted in advance.

Such a simple approach leaves a large space for the adversary to perform steganalysis. Fridrich and Goljan [3] have surveyed several methods in analysing LSB substitution. One of those is known as histogram analysis [1][2]. This method plots a histogram with the pixels in the testing image. As shown in Figure 1, if an image is embedded by LSB substitution, the histogram of the image will be changed in a “pair-wise” way. These pair-wise blocks are also known as *Pairs of Values* (PoV) [2]. This effect can be identified by applying the χ^2 -test [4] and the adversary can justify whether the testing image is embedded with LSB substitution.

LSB matching [5] have a subtle difference from LSB substitution by the following: If the embedded data

have the same value as the LSB of a particular pixel, the pixel will be left unchanged. Otherwise, the pixel will be randomly added or subtracted by 1. The pair-wise effect of PoVs will become less significant in this algorithm. However, as we will show in the later section, the histogram will still be affected by this algorithm. That also means histogram analysis still works on this algorithm.

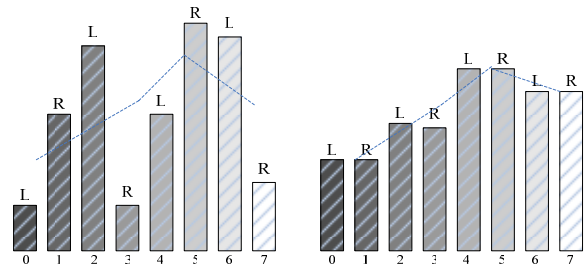


Figure 1. Histogram of an image before and after being loaded by LSB substitution algorithm. (adopted from [2])

Our research is motivated to designing an image data hiding scheme *without* changing the histogram of the hosting image. This kind of data hiding scheme will be perfectly imperceptible to histogram analysis attack. We also aware that the distortion introduced by the scheme must be well controlled. Otherwise, it will be visually detectable by a human.

The paper is organized as follows. In section 2 we will propose our scheme. To evaluate our contribution,

we will describe how the experiments were conducted and present the experimental results in section 3. It is followed by the analysis of the experimental results in section 4. In section 5, we will raise some discussions about the scheme. The paper is concluded in section 6.

2 The Proposed Scheme

In this section, we describe our proposed scheme which includes two phases: (1) Rearrangement (2) Swapping.

2.1 Rearrangement

Let G be the host image with $m \times m$ pixels and 256 gray levels. We label the pixels from top to down, from left to right. These pixels are further divided into 256 groups, denoted by V_0, V_1, \dots, V_{255} , according to their value. Every two groups V_{2i} and V_{2i+1} are paired up where i is from 0 to 127. The size of each group will be recorded. For example, the value of the 13th and 34th pixels are n and the 7th, 14th, 18th, 24th, 48th are $n+1$, we will have $V_n = \{13, 34\}$ and $V_{n+1} = \{7, 14, 18, 24, 48\}$ and the size of V_n is 2 and the size of V_{n+1} is 5, as shown the left most blocks in Figure 2.

Now, for every pair of groups V_n and V_{n+1} , we merge and sort the pixels within the two groups. Thus we have the sequence $T = \{7, 13, 14, 18, 24, 34, 48\}$ as an example. The pixels in the sequence T will be reassigned to V_n and V_{n+1} in alternating order without changing the size of the groups. As an example shown in Figure 2: the 1st element 7 is assigned to V_n , the 2nd element 13 is assigned to V_{n+1} , and so on.

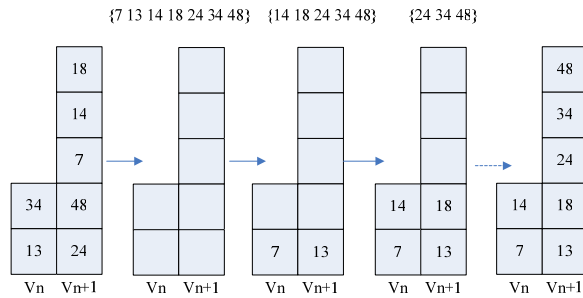


Figure 2. An example demonstrates the Rearrangement phase

2.2 Swapping

After the rearrangement phase, each pair of groups V_n and V_{n+1} has the following property: suppose the smaller size of the pair of the groups is k , there are k exclusive pairs of pixels that one with smaller label is in V_n and the one with larger label is in V_{n+1} . As in the example, k is 2 accordingly and we have 2 exclusive pairs $\{7, 13\}$ and $\{14, 18\}$ that 7 and 14 are in V_n where 13 and 18 are in V_{n+1} .

We make use of this property to embed the secret. We first identify all the exclusive pairs from all the pairs

of groups. If we wish to be embedded the secret bit 0, we do nothing about the exclusive pair. If secret bit 1 is going to embed, we swap the position of the elements in the exclusive pair. An example is demonstrated in Figure 3.

After embedding all the secret bits, we have a new set of groups $\{V'_0, V'_1, \dots, V'_{255}\}$. Now, all the pixels are updated according to which group are they belongs to. This completes the embedding process.

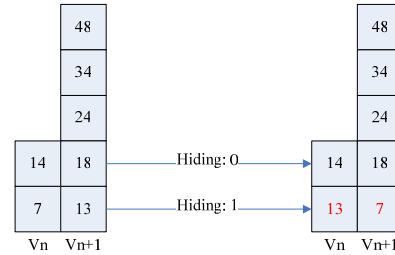


Figure 3. An example demonstrates the Swapping phase

2.3 Data Extraction

Upon receiving the stego-image, the receiver extracts the secret bits stream by building group $\{V'_0, V'_1, \dots, V'_{255}\}$ and sorts the pixels in ascending order according to their label in each group V'_i . Then, these groups are paired up. Exclusive pairs are identified by cutting the groups horizontally. As shown in the right hand side of Figure 3, we have the exclusive pairs $\{13, 7\}$ and $\{14, 18\}$. We then figure out that the information hidden is 1 and 0, respectively. This can be figured out by the reverse order of $\{13, 7\}$ and the proper order of $\{14, 18\}$.

3 Experimental Results

To evaluate our scheme, several experiments are designed. The result will be compared with LSB substitution and LSB matching. The first test is on the distortion and the data hiding capacity. The second test is on the vulnerability of histogram analysis. The third test is a visual test proposed in [2].

3.1 Distortion and Capacity Test

In this test, the images Lenna (Figure 4a), F16, and Pepper (Figure 7a, 7b) which have the size of 512×512 and 256 gray levels, will be used as the host image for the three algorithms with the same random sequence. Since the capacity of our algorithm is upper bounded by 0.5 bits/pixel, the other two algorithm will embed the same amount of data for comparisons. The images outputted by the three algorithms will be measured through the distortion they have brought in. The distortion is measured by the value PSNR, as follows:

$$MSE = \frac{1}{m^2} \sum_{i=1}^m \sum_{j=1}^m (\alpha_{ij} - \beta_{ij})^2 \quad (1)$$

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \quad (2)$$

The result of this test will be tabled in Table 1. A large PSNR means small distortion that the output image makes. Also, the stego-images of Lena, produced by the three algorithms, will be shown in Figure 4.

3.2 Histogram Test

In this test, we plot the histogram of the outputted images and calculate the divergence [6] between each output image and the hosting image. Divergence is a natural distance measure from a “true” probability distribution P to an arbitrary probability distribution Q . It is calculated as follows:

$$\text{Divergence} = \sum_{x=0}^{255} p(x) \log \frac{p(x)}{q(x)} \quad (3)$$

Where, $p(x)$ and $q(x)$ are the probability of a pixels having a grey level x of the output image and hosting

image respectively. Since, $p(x)$ and $q(x)$ might be 0 for some value x , we add a negligible terms to every $p(x)$ and $q(x)$ to avoid division by 0 error and $\log 0$ error.

The result of this test will be tabled in Table 1. A larger value of divergence means the histogram being distorted more severe. Also, the histograms of the stego-images and the hosting image are plotted in Figure 5.

3.3 Visual Test

As suggested in [2], if we draw the LSB of an image without embedding secret, it will looks like Figure 6a. If it is embedded with LSB substitution, it would be completely noisy. This test, however, may not be able to handle noisy images or highly textured images from stego-images. Also it is hard to automatize and their reliability is highly questionable. Nevertheless, this visual test will still be performed to justify if our scheme is easily being detected.

The image Pepper will be used for this test. The results of this test are given in Figure 6.



Figure 4. From left to right, (a) unaltered Lena, (b) LSB substitution, (c) LSB matching, (d) our scheme

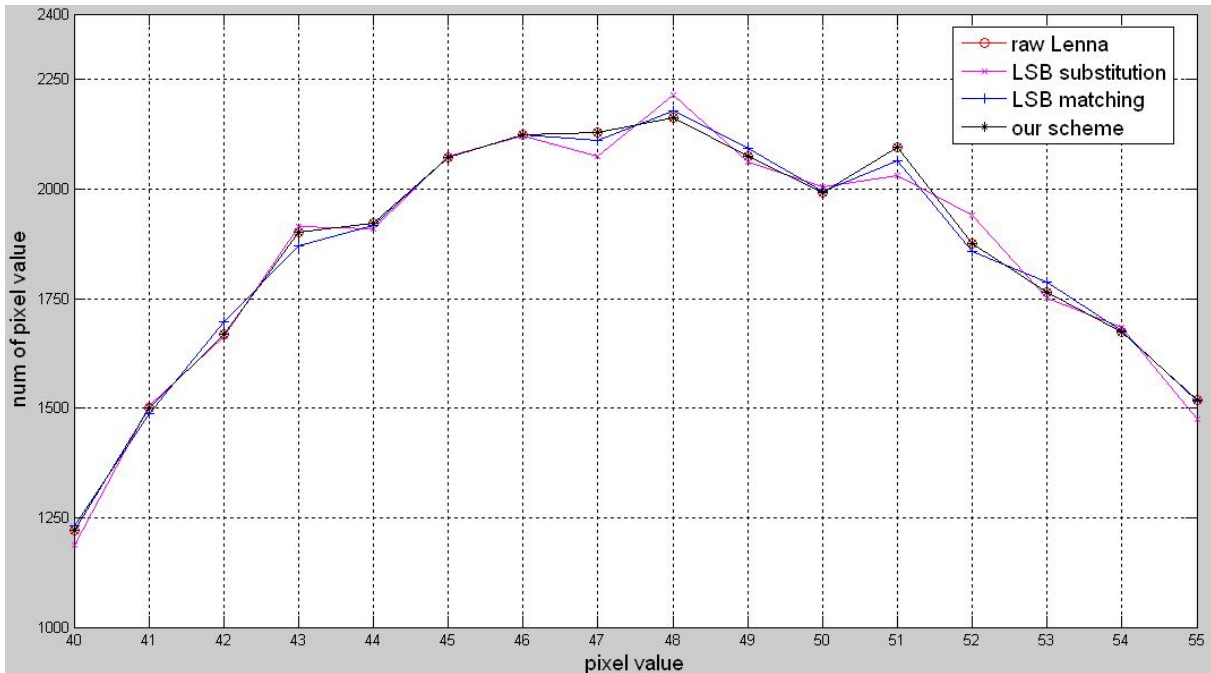


Figure 5. Histogram plot of hosting image and stego-images

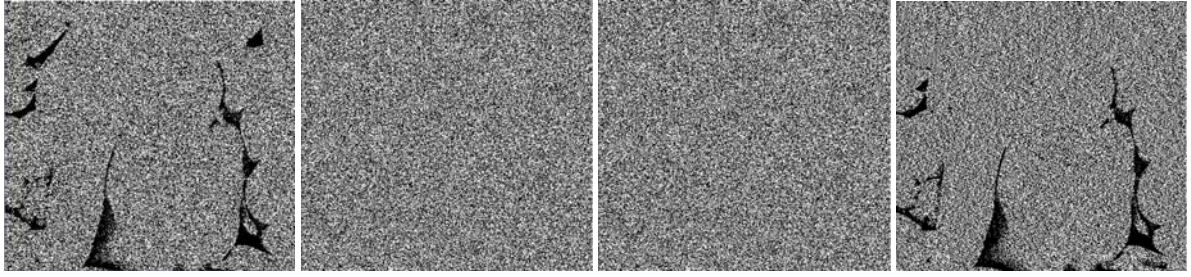


Figure 6. LSB draw of, from left to right, (a) hosting image, (b) LSB substitution, (c) LSB matching, and (d) our scheme.



Figure 7. From left to right, (a) F16 and (b) Pepper

Table 1. Experimental Result of the Distortion and Capacity Test, and Histogram Test

Lena	PSNR (dB)	Divergence (bit)	Capacity (bit/Pixel)
Our scheme	51.152	0	0.4838
LSB substitution	54.314	0.000924	0.4838
	51.161	0.001826	1
LSB matching	54.314	0.000597	0.4838
	51.161	0.001121	1
F16	PSNR (dB)	Divergence (bit)	Capacity (bit/Pixel)
Our scheme	51.187	0	0.4760
LSB substitution	54.371	0.003915	0.4760
	51.143	0.012294	1
LSB matching	54.371	0.003496	0.4760
	51.143	0.009648	1
Pepper	PSNR (dB)	Divergence (bit)	Capacity (bit/Pixel)
Our scheme	51.354	0	0.4593
LSB substitution	54.538	0.018789	0.4593
	51.151	0.091965	1
LSB matching	54.538	0.019131	0.4593
	51.151	0.095241	1

4 Analysis

As we can see from the Table 1, our hiding capacity and PSNR are relatively low, compared to the existing schemes. And, the hiding capacity of our algorithm depends on the histogram of the hosting image. In the worst case, for example, $V_{2i} = 0$ for all i , the hiding capacity is 0 bit/pixel. In the best case, for example, all V_i are the same, the hiding capacity would be 0.5 bit/pixel. Fortunately, the hiding capacity from the experiment shows that we have average hiding capacity of 0.473 bit/pixel. So, for a 512×512 hosting images, we can hide 121 Kb in average.

We generally believe image with PSNR value higher than 32db is imperceptible to human eyes that it is loaded with secret information. Experimental results show our scheme works far beyond this threshold.

From Figure 5, we learn that LSB substitution and LSB matching yield different histogram profiles from the hosting image. The histogram produced by our scheme perfectly collides with the one of hosting image. This property makes our scheme perfectly secure against histogram analysis attack. The divergence given in Table 1 concurs with Figure 5.

The visual tests shown in Figure 6 justify our scheme is secure against simple visual test. It might be evidence that our scheme is much more secure than the other two schemes.

5 Discussion

Although our scheme is secure against histogram analysis, it does not mean it is secure enough to be imperceptible by any other algorithm. Readers may find out that every pair of groups produced by our

scheme behaves as follows: If the size of the groups, for example, V_n and V_{n+1} are not equal, let say V_n is smaller, then, all the unpaired labels in V_{n+1} are larger than the element in V_n . In the example we illustrated in section 3, the unpaired labels in V_{n+1} {24, 34, 48} larger than all the labels in V_n {13, 14}. If there are many pairs of groups behave in this way, the adversary has a high confidence that our scheme has applied in this stego-image.

To avoid this simple detection, instead of label the pixel in Top-bottom-left-right order, we should randomly permute them with a private key, which is shared by the sender and receiver only.

Since this is a new scheme, there is no existing steganalysis regarding to this scheme. To the best of our knowledge, the most efficient steganalysis for LSB substitution and LSB matching are of *Dual Statistics* methods [3][5]. It is much easier to prove a scheme is vulnerable to some attack than to prove it is secure against it. In the future, we try to prove that our scheme is also secure against this stream of attacking methods by looking it from a statistics point of view.

6 Conclusion

In this paper we have proposed an image data hiding scheme for gray level image that is perfectly imperceptible by histogram analysis. By scarifying data hiding capacity and distortion level, our scheme achieves a better security level than LSB substitution and LSB matching in histogram analysis and visual test.

Acknowledgements

The authors wish to acknowledge the anonymous reviewers for valuable comments. This research was supported in part by the National Science Council, Taiwan, under contract NSC 95-2918-I-007-014 and NSC 95-2221-E-007-021.

References

- [1] N. Provos, "Defending Against Statistical Steganalysis", 10th USENIX Security Symposium, Washington, DC, 2001.
- [2] A. Westfeld and A. Pfitzmann, "Attacks on Steganographic Systems," Lecture Notes in Computer Science, vol.1768, Springer-Verlag, Berlin, 2000, pp. 61–75.
- [3] J. Fridrich and M. Goljan, "Practical steganalysis of digital images—State of the art," in Proc. SPIE Security Watermarking Multimedia Contents, vol. 4675, E. J. Delp III and P. W. Wong, Eds., 2002, pp. 1–13.
- [4] W. Dixon, F. Massey: Introduction to Statistical Analysis. McGraw-Hill Book Company, Inc., New York 1957.
- [5] A. Ker, "Steganalysis of LSB Matching in Grayscale Images." IEEE Signal Processing Letters, vol. 12(6), pp. 441–444, 2005.
- [6] S. Kullback and R. A. Leibler, "On information and sufficiency," Annals of Mathematical Statistics, vol. 22, pp. 79–86, 1951.

Chromatic Variance Prediction

Robert N. Grant¹, and Richard D. Green²

University of Canterbury, Dept. Computer Science, New Zealand.

¹robert.grant@canterbury.ac.nz, ²richard.green@canterbury.ac.nz

Abstract

In the area of vision-based local environment mapping, inconsistent lighting can interfere with a robust system. The HLS colour model can be useful when working with varying illumination as it tries to separate illumination levels from hue. This means that using hue information can result in an image invariant to illumination. This can be valuable when trying to determine object boundaries, object identification and image correspondence. The problem is that noise is greater at lower illumination levels. While removing the illumination effects on the image, separating out hue means that the noise effects of non-optimal illumination remain. This paper looks at how the known illumination information of pixels can be used to accurately predict and reduce noise in the hue obtained in video from a colour digital camera.

Keywords: Hue noise, computer vision, illumination invariance

1 Introduction

With vision-based local environment mapping consistency in the environment is highly desirable. This includes consistent lighting conditions which means that most research is conducted under as controlled an environment as possible. Unfortunately this is not a luxury that can be afforded in real world applications which means that many projects can not achieve widespread public use. The problem is that illumination in general usage is unpredictable, causing tasks such as colour tracking for object recognition to be problematic because the intrinsic characteristics of digital cameras causes the value of hue to vary with illumination. There have been projects in the past that have tried to track the colour of an object as it changes with varying levels of success [1][2][3] shown in figure 1. While these methods can work, they often need to be reinitialised if tracking is lost and are computationally inefficient leaving less for the primary vision application.

This research takes the approach of an illumination invariant filter on video data, acquiring video frames and converting them into a normalised illumination format consisting of the raw colours of the scene. Conversion to the HLS colour model shown in figure 2 is the starting point to this transformation as the hue component of this colour model is essentially the colour of an object with the illumination intensity information stripped out. White balancing is also necessary to remove light source colouring effects on objects.

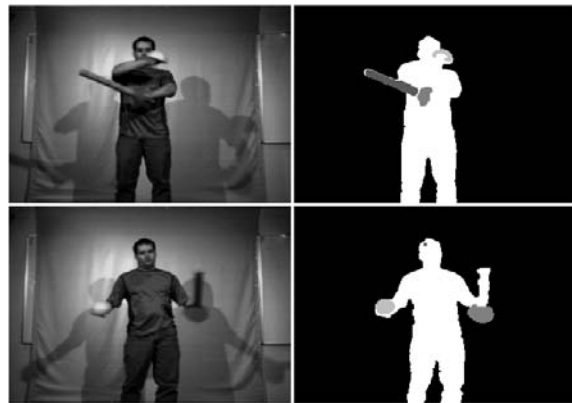


Figure 1: Frames from a dynamic colour tracker.

This would be an ideal illumination invariant input for a computer vision system as with accurate white balancing hue values of objects should stay consistent under any light source. Unfortunately this is not the case. When lighting decreases, understandably noise increases. This means any darker areas of a frame of video result in noisier hue information than lighter areas. To counter this noise, first a reliable predictor of it needs to be found. This research is investigating whether or not the other two components of the HLS colour model (luminance and saturation) are accurate predictors of hue variance with changing levels of illumination.

2 Related Work

There is much research into robot environment mapping [4][5][6] but most completely ignore how

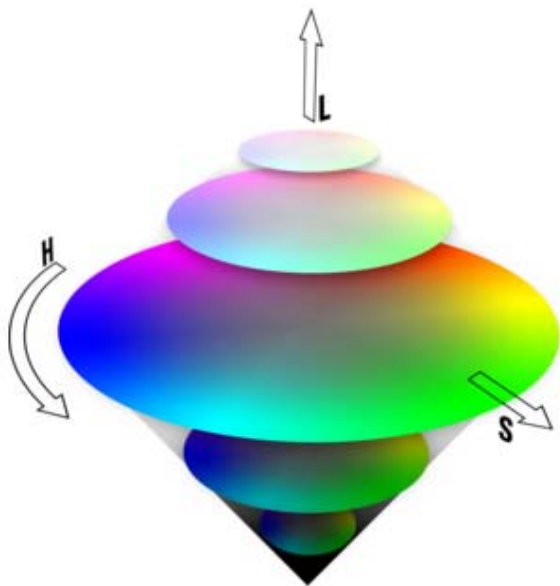


Figure 2: The HLS colour model.

their system would function under low lighting conditions. Because noise increases in areas of low illumination, reliably identifying colour regions or previous points of interest drops significantly. Some research is focused on operating in unlit environments using alternative sensing such as ultrasound, radar, laser scanning or infra-red [7].

Research into shadow detection and removal attempts to locate shadows so that they can be isolated from the objects they belong to. Yao and Zhang [8] present a shadow detection algorithm which uses sample images of shadow and non-shadow to create a histogram using the luminance and chroma information. It then uses the histogram to predict the shadow regions in the image. This method assumes the uniformity of shadows and lighting conditions and is shown in figure 3. Wang *et al* [9] removes shadows from object detection of cars with a combination of static background subtraction and foreground to background comparisons to differentiate shadow pixels from object pixels.

Salvador *et al* [10] use an invariant colour model to separate luminance from chrominance information. They can then segment the object from the background by restricting the scene to a uniformly coloured background and object. This meant that the background and object could be identified using the invariant chrominance and luminance could be used to identify shadowed and non-shadowed areas. The use of an invariant colour model to segment objects is useful because it allows for light intensity changes while still maintaining reliable segmentation. An example of the segmentation is



Figure 3: Two different shadow detection algorithm results.

shown in figure 4. It allows us to ignore shadowing and other lighting effects altogether.

Invariant colour models become unreliable when the light sources in a scene are not 'white'. For example, when a red object is under a slightly yellow light source it will appear slightly orange with regions shadowed from the yellow light source appearing more red. White balancing is a process used to correct the effects of discoloured lighting in an image [11].

By implementing reliable white balancing and using an invariant colour model an object's colour should rarely change due to illumination changes. Unfortunately this is not the case when the intensity of light reflected from an object nears the outer limits of the camera's visible range. Cameras are not sensitive to these areas and so noise causes the colour/hue of an object to vary dramatically.

3 Experimentation

3.1 Method

The following experiment aims to discover the correlation between the effect of noise levels on hue and the levels of the other two components of HLS colour (luminance and saturation). These may be useful predictors as they represent the amounts of light coming into the camera and an indication of the accuracy of hue. A predictable correlation between these two factors enables countering these noise effects.

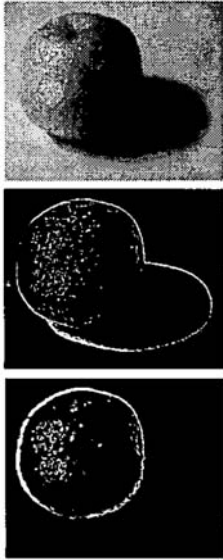


Figure 4: Shadow segmentation using an invariant colour model.

Different scenes were selected for range of colour and brightness. The camera exposure period was up to 30 seconds to collect accurate HLS colour information for each pixel of the scene over time. Each pixel was classified by means of averaging to a specific luminance and saturation pair. Standard deviation of the hue values collected were calculated for each pixel. Hue variances are added to an array of minimum hue variance for each luminance by saturation pair (256x256).

3.2 Results

Figure 5(a) shows the data extracted from this experiment. It can be seen that at low and high luminance and low saturation values with results indicating that the amount of noise in hue can spike significantly. There are also some scattered hue noise peaks in the data as can be seen in the graph. These can be attributed to other effects caused by the method of data collection. With a near stationary camera, tiny movements can cause large changes in pixel colour near the edges of objects. because of this the lowest variance is always selected when duplicate luminance and saturation pairs arise, this helps reduce these effects.

These results suggest that it is possible to correctly predict the noise of a pixel without any temporal information. Luminance and saturation may therefore be accurate predictors of the hue variance for any given pixel.

3.3 Curve fitting

From the data found in figure 5(a) we can see that, with the correct formula, hue variation can

be predicted. The problem is finding this fitting a mathematical curve to this data. By analysing a cross section of the data along one axis at a time, it was found that both axes closely fit an inverse squared curve which when multiplied together produced a close fit to the data. In the case of the luminance direction the symmetry means the term L is inverted half way.

When $L < 128$:

$$H = \left(\frac{\alpha}{S^2} + \beta\right) \times \left(\frac{\gamma}{L^2} + \delta\right)$$

Else:

$$H = \left(\frac{\alpha}{S^2} + \beta\right) \times \left(\frac{\gamma}{(255 - L)^2} + \delta\right)$$

To match the data from the previous experiment, the coefficients found to be a close fit were: $\alpha = 2913$, $\beta = 1.18$, $\gamma = 1974$, $\delta = 0.6301$. This produces the predicted graph in figure 5(b). These coefficients would be different for different cameras but the equation should still be the same. Each camera would need to be calibrated for a specific noise curve. Figure 6 shows how this can be applied to a frame of video (a) resulting in image (d) with noisy hue areas having less of an effect.

4 Conclusions

This research is working towards creating an illumination invariance filter for colour camera input. This can be used to better identify or correlate objects regardless of changes in lighting conditions or viewing angle. While white balancing and illumination invariant colour models are on the way to achieving this, they come across large amounts of noise when trying to identify colours that have intensities outside of the sensitive range of the camera. This research has remedied this by showing that an equation can be used to predict how reliable colour values are across an image. In this way correlation between a persistent representation and the camera input can be made more reliably.

References

- [1] R. N. Grant and R. D. Green, "Tracking colour movement through colour space for real time human motion capture to drive an avatar," in *Proceedings of Image and Vision Computing New Zealand*, Nov 2004.
- [2] K. Nummiaro, E. Koller-Meier, and L. J. V. Gool, "Object tracking with an adaptive color-based particle filter," in *Annual Symposium of the German Association for Pattern Recognition*, p. 353 ff., 2002.

- [3] J. Vergs-Llah, J. Aranda, and A. Sanfeliu, "Object tracking system using colour histograms," in *Proceedings of the 9th Spanish Symposium on Pattern Recognition and Image Analysis*, pp. 225–230, May 2001.
- [4] K. Fintzel, R. Bendahan, C. Vestri, S. Bougnoux, S. Yamamoto, and T. Kakinami, "3d vision system for vehicles," in *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, pp. 174–179, June 2003.
- [5] J. Takeno and S. Hachiyama, "A collision-avoidance robot mounting ldm stereo vision," in *Proceedings of the IEEE International Conference on Robotics and Automation, 1992.*, vol. 2, pp. 1740–1752, May 1992.
- [6] M. Tomono, "3-d localization and mapping using a single camera based on structure-from-motion with automatic baseline selection," in *IEEE International Conference on Robotics and Automation*, pp. 3342–3347, April.
- [7] T. Tsuji, H. Hattori, M. Watanabe, and N. Nagaoka, "Development of night-vision system," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, pp. 203–209, Sept 2002.
- [8] J. Yao and Z. Zhang, "Systematic static shadow detection," in *International Conference on IPattern Recognition*, vol. 2, pp. 76–79, Aug 2004.
- [9] J. M. Wang, Y. C. Chung, C. L. Chang, and S. W. Chen, "Shadow detection and removal for traffic images," in *IEEE International Conference on Networking, Sensing and Control*, vol. 1, pp. 649–654, Mar 2004.
- [10] E. Salvador, A. Cavallaro, and T. Ebrahimi, "Shadow identification and classification using invariant color models," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 1545–1548, May 2001.
- [11] H.-K. Lam, O. C. Au, and C.-W. Wong, "Automatic white balancing using standard deviation of rgb components," in *International Symposium on Circuits and Systems*, vol. 3, pp. 921–924, May 2004.

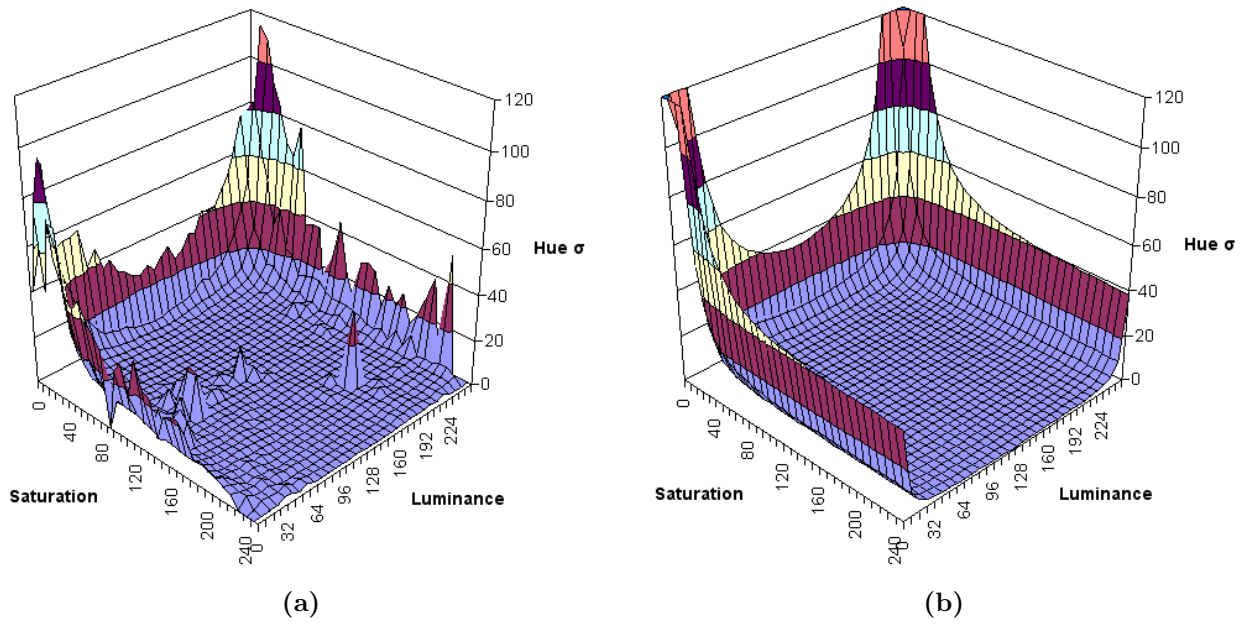


Figure 5: Hue standard deviation vs. Saturation and Luminance (a) Experimental results (b) Predicted curve.

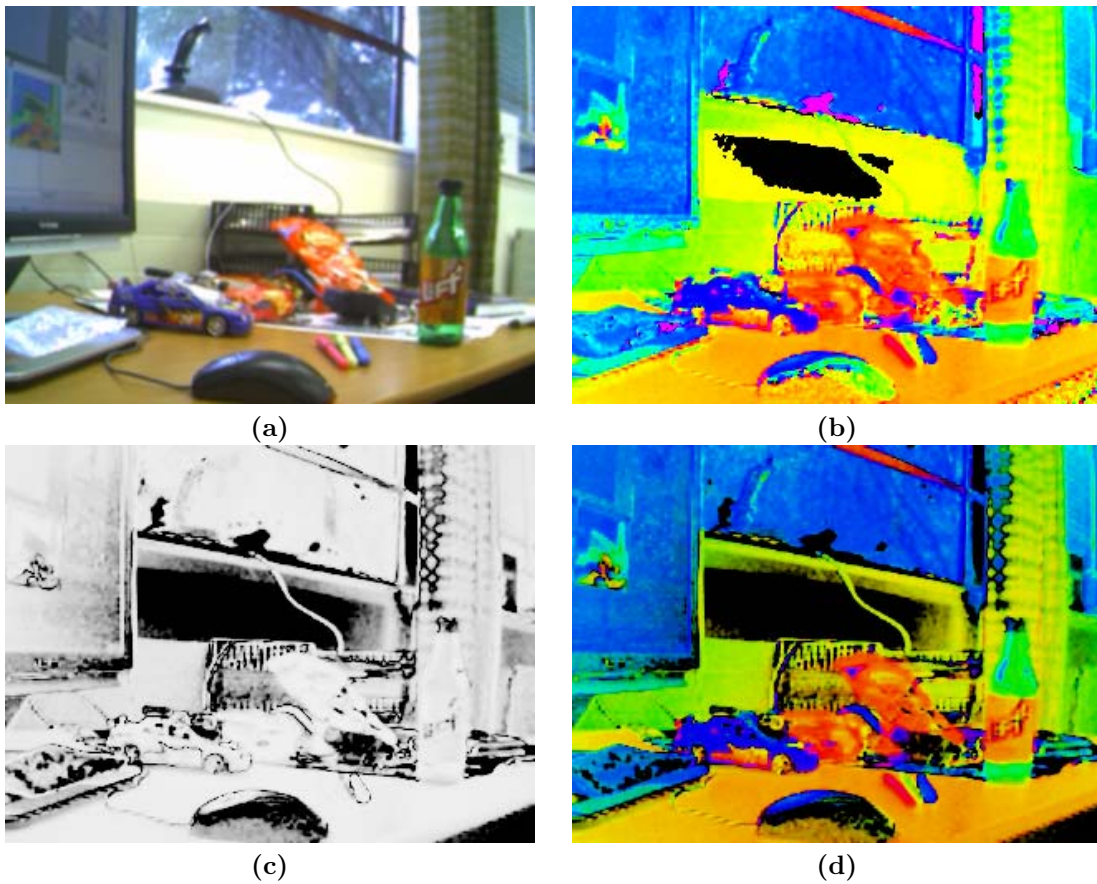


Figure 6: (a) Original image (b) Hue image (c) Predicted hue noise image (d) Hue image with darkened noisy areas.

Near optimal non-uniform interpolation for image super-resolution from multiple images

A. Gilman, D.G. Bailey

Institute of Information Sciences and Technology, Massey University, Palmerston North

Email: a.gilman@massey.ac.nz

Abstract

Non-uniform interpolation is a common procedure in image processing. A linear interpolation filter is generally a weighted combination of the inputs. Optimal filter coefficients (in a least squares sense) can be derived if the interpolated image is known beforehand. The weights of a general linear interpolation filter are independent of content and only depend on the relative positions of the available samples. The optimal coefficients are shown to be relatively independent of the content experimentally in absence and presence of noise, allowing non-uniform images to be interpolated using coefficients that have been optimised on a synthetic image. This results in a linear interpolation with computational complexity of the same order as nearest neighbour or bilinear, but with a near optimal performance.

Keywords: Image super-resolution, non-uniform interpolation, scattered interpolation.

1 Introduction

Non-uniform interpolation, also known as scattered interpolation, is a key step in image super-resolution from multiple images [1]. After the input low-resolution (LR) images are registered to the high-resolution (HR) grid, these can be combined (with appropriate offsets) to form a compound non-uniformly sampled image. An interpolation procedure can be applied to resample the compound image at the uniform positions of the high-resolution grid. There are a large number of interpolation methods exist, each making assumptions about the surface of the image function. The choice of a method depends strongly on the application specific requirements, as there is a trade-off between computational complexity, memory requirements, and optimality of the result. Sensitivity to the accuracy of registration procedure can also play a role in the selection.

The scope of this work was to look at global translational motion only, with low noise levels. Image degradation was assumed to be due to the camera point spread function only, constant in time and linear space-invariant for all input images.

Because of the assumption of global translational motion only, the non-uniform compound image is actually semi-uniform. Of course in this case the generalised sampling theorem [2,3] can be applied to reconstruct the exact high-resolution image, as long as average sampling rate is above the Nyquist rate. However, this procedure is computationally expensive and is sensitive to even low levels of noise. Our main interest is super-resolution methods with low

computational complexity; therefore we consider this approach unsuitable.

The remainder of this article summarises a number of interpolation methods that can be implemented as digital filters, and develops a new, near optimal, method for computing the weights of a linear interpolation filter.

2 Near optimal interpolation

The simplest method for image interpolation is nearest neighbour interpolation [4]. For each point on the HR grid, the closest known LR pixel is selected and the value of that pixel is simply used as the value at the grid point. This method, therefore, implicitly assumes a piece-wise constant model for the image. It is the fastest of all interpolation methods as it considers only a single pixel – that closest to the grid point being interpolated.

Another simple and well-established method is bilinear interpolation. It can be applied to super-resolution in the following way [5]. To interpolate a given point on the HR grid, the closest LR pixel is used, along with its three neighbours from the same LR image, as pictured in figure 1. These three pixels are picked so that they are the next closest pixels to the point being interpolated. The four LR pixels are the vertices of a square, with the point being interpolated located inside the square. The value at the point is computed using a bilinear weighed sum of the four vertices. Bicubic interpolation can be implemented in a similar way, selecting 16 closest points and applying bicubic weights.

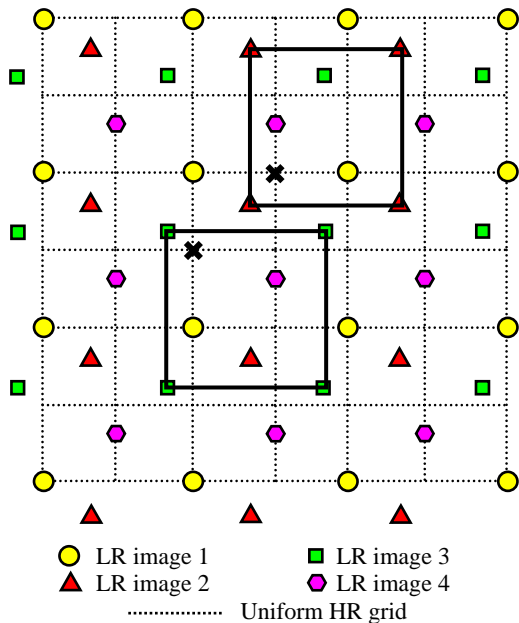


Figure 1: Increasing resolution by a factor of two, using bilinear interpolation. The four input samples used are generally not the closest samples available.

While this is fine for a single image, where there are many low-resolution images, three (or 15 in the case of bicubic) of the pixels that are used are not necessarily the closest, as these may be in other input images. Logically, the closest inputs to the output pixel are more likely to contain relevant information about the HR pixel value.

Instead of using four closest pixels from a single input image, the selection can be made from all pixels from all input images. A problem then arises of determining the weights for each of these inputs.

One approach is to determine the weights simply by a function of distance of each input sample from the output point (i.e. inverse distance, inverse distance squared, etc). This method has first been attributed to Shepard [6].

An alternative is to use Delaunay triangulation and then fit a plane or other type of surface to each triangle to interpolate an HR grid point inside that triangle [7].

Other methods also exist and can be generalised as follows. First, N input pixels around the point we want to interpolate are selected (or all input pixels falling inside a radius r around the output point). Then, the value for the output point is calculated as a weighted combination of these known pixels. The weights depend on the interpolation method. Since different methods would produce different weights, the obvious question is “which weights are the best?”

If the ideal HR image is known (i.e. ground truth), the weights that minimise the squared error could be computed. Such weights would be optimal in a least squares sense. The problem is that the desired output

image is unknown; otherwise there would be no point to interpolate.

The weights of a general linear interpolation filter do not depend on the image content, but on the relative positions of the available samples. Therefore, it is hypothesised that the optimal weights should depend only weakly on the actual image content. If this is the case, then the optimal weights derived from one image should be close to optimal (thus near optimal) on other images with the same offsets. Hence, a synthetic image can be used to derive the weights, which are then applied to the input images.

In terms of implementation, such a method can be implemented as a two-dimensional finite impulse response filter, just as all the other methods described in this section. Hence, all these methods should be of similar computational complexity, apart from the overhead of calculating the coefficients for the “near optimal” method, which is run just once before the input images are processed.

3 Results and discussion

3.1 Experimental setup

To assess the performance of the above methods, it is necessary to have a ground truth high-resolution image. If the low-resolution images are simply captured, the ground truth is unknown. Hence we used a method similar to Bailey [8] to generate a number of LR images from a single very high-resolution image through a simple imaging model. Image ‘beach’ (as pictured in figure 2) was selected to be the test image – it has dimensions of 1700×1700 pixels. To form LR images, the source image was filtered using a 20×20 square box average filter to simulate area integration, shifted by random (integer pixel) offsets, then down-sampled by a factor of 20. Finally, Gaussian noise was added to simulate the effects of various noise sources within the process. To form the HR image, the source image was filtered using the same filter, but down-sampled only by a factor of 10. Note that the high-resolution image is blurred to the same degree as the low-resolution images. This enables the performance of interpolation methods alone to be investigated, hence deblurring is left out. In addition, the exact known offsets were used to ensure there is no misregistration of the low resolution images.



Figure 2: Image ‘beach’.

The down-sampling procedure resulted in offsets that are integer multiples of 0.05 of a pixel. Four LR images were used to super-resolve a single HR image by a factor of two. Simulating every possible combination would be very time consuming; hence, a Monte Carlo simulation was used consisting of 10,000 runs with randomly selected offsets. On each run the four randomly-offset low resolution images were combined into a single super-resolved image using one of the interpolation methods. The output image was compared to the ground truth image and the mean square error was calculated.

3.2 Experimental results

The results can be interpreted using an inverse cumulative distribution function (iCDF), also known as percent point function or quantile function [9]. To form this function, the errors from all the runs are ranked in ascending order and plotted, with the probability of 0.0001 associated with the smallest error and probability of 1.0 associated with the largest error. The probability, therefore, gives the probability that the interpolation error is less than the associated error.

The inverse cumulative distribution functions of the errors for nearest neighbour, bilinear and optimal interpolations can be seen in figure 3. This plot shows the expected performance for a particular percentile. So for example for 50th percentile (the median), nearest neighbour interpolation is expected to yield MSE below 16.4×10^{-5} .

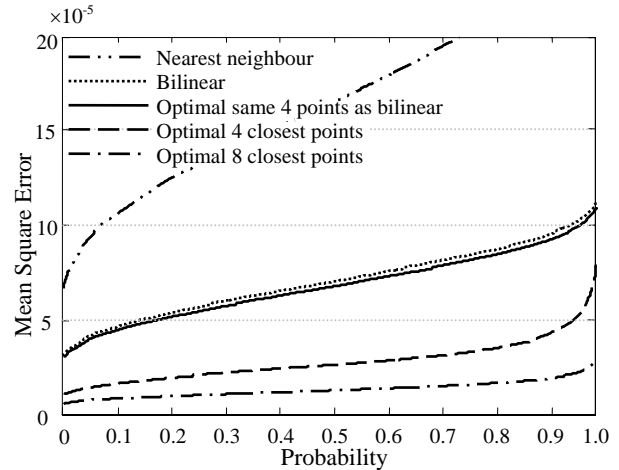


Figure 3: Inverse cumulative distribution functions (percent point functions) for nearest neighbour, bilinear, and optimal interpolations of image Beach.

It can be seen that the bilinear interpolation performs significantly better than the nearest neighbour interpolation. This result is intuitive, as bilinear interpolation uses more input information (four input points in comparison to one input point). The line just under bilinear corresponds to the optimal result that could be achieved if the same four points as used by the bilinear are utilised. For a given image, and a selection of input pixels used to perform the interpolation, this optimal method will give the smallest mean squared error (MSE) that can be obtained using a linear interpolation filter and can be used as a benchmark to compare other methods. The proximity of bilinear to optimal result is a good indication that bilinear interpolation is always stable and utilises the input information extremely well.

However, as already mentioned, the four input points used by the bilinear are not the ideal choice for super-resolution. Points further away from the desired location carry less relevant information, so ideally we want to use closest possible input points. This is confirmed by figure 3: optimal interpolation using four closest points yields a factor of two improvement over using the same points as the bilinear interpolation.

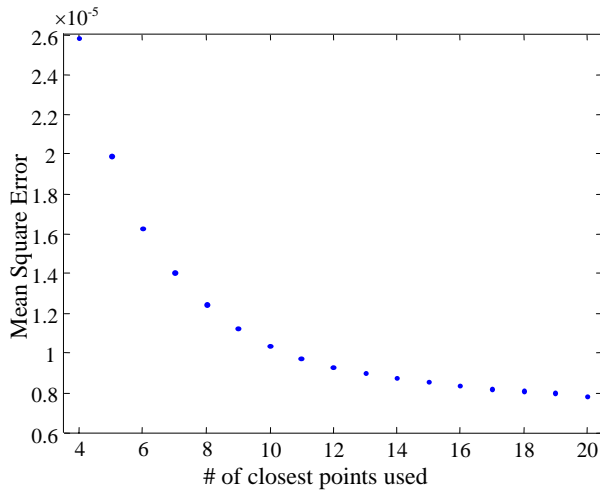


Figure 4: Images ‘sleep’ and ‘disk’.

Figure 4 shows the optimal performance (median of iCDF) using between 4 and 20 closest points. It is clear that although the performance improves with an increasing number of points, the gains from using additional points decreases with points further away from the point being interpolated. Between eight and ten points seem to be a good compromise between accuracy and computational effort.

Two very different images were used to simulate the near optimal coefficients to test the hypothesis that coefficients derived from one image should work on another image. These are pictured in figure 5.

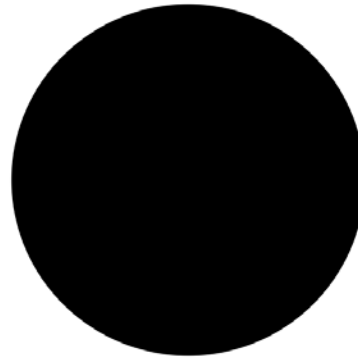


Figure 5: Images ‘sleep’ and ‘disk’.

Image ‘sleep’ was chosen because it differs from image ‘beach’, but has similar statistics. Image ‘disk’ was chosen because it has significantly different statistics to have some idea how image content affects the results. It is also a synthetic image that can be used to calculate the coefficients for an arbitrary image.

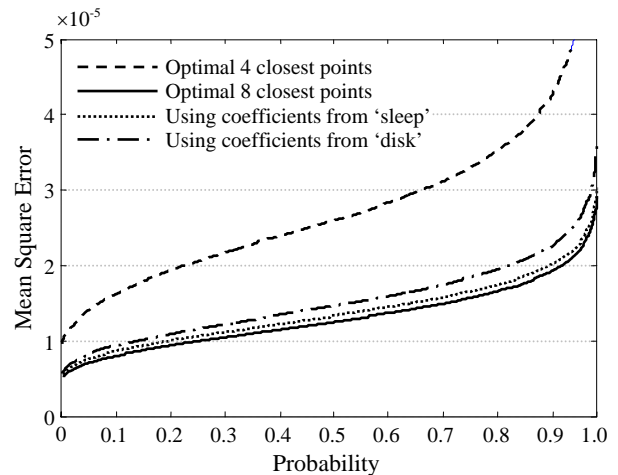


Figure 6: iCDFs of optimal interpolation and interpolation using coefficients optimised on images ‘sleep’ and ‘disk’ (using 8 closest points) and optimal using 4 closest points for comparison.

Eight closest points were chosen to be used to test the hypothesis. Figure 6 shows that using the coefficients optimised on images ‘sleep’ and ‘disk’ generate results very close to that of optimal. Image ‘disk’ has very different statistics, but it contains edges of all directions. This is possibly the reason why coefficients optimised on it offer reasonable interpolation results. These results are tabulated in Table 1.

Table 1: 1st, 2nd, and 3rd quartiles of iCDFs plotted in figure 6.

Method (8 points)	1 st quartile ($\times 10^{-5}$)	2 nd quartile ($\times 10^{-5}$)	3 rd quartile ($\times 10^{-5}$)
Optimal	0.99 (100%)	1.25 (100%)	1.56 (100%)
Coefficients from 'sleep'	1.06 (107%)	1.32 (106%)	1.64 (105%)
Coefficients from 'disk'	1.15 (116%)	1.45 (116%)	1.83 (117%)

The previous experiment was performed in the absence of noise. This is seldom true in practical imaging systems. Even if most sources of noise are minimised, there is still quantisation noise. For a typical 8-bit system this would have a standard deviation of 0.29 of a greyscale level. To check whether noise has any significant effect on the predicted coefficients, we super-resolved the 'beach' image at different levels of additive white Gaussian noise using coefficients optimised on the same image but without noise.

Table 2: Interpolating 'beach' at different noise levels using coefficients optimised on the same image at those noise levels and coefficients optimised without noise.

Median MSE ($\times 10^{-5}$) / Noise s.d.	Optimised at that noise level	Optimised with no noise
0	1.25	1.25
0.5	1.76	1.90
1	2.67	3.58
2	4.90	10.4
3	7.35	21.9
4	9.91	37.8

Second column of Table 2 shows the performance of the optimal coefficients at various levels of noise. As expected, the MSE increases with the noise standard deviation. Third column shows the performance of coefficients optimised without noise. It can be seen that the relative performance deteriorates as the noise is increased.

Addition of Gaussian noise to the LR image formation model is investigated as a possible way of improving performance with noisy inputs. Five sets of coefficients are created, each optimised on image 'beach' using noise levels of zero, one, two, three, and four. Each set of coefficients is applied to inputs with various levels of noise (between zero and 4) and the results are plotted in figure 7.

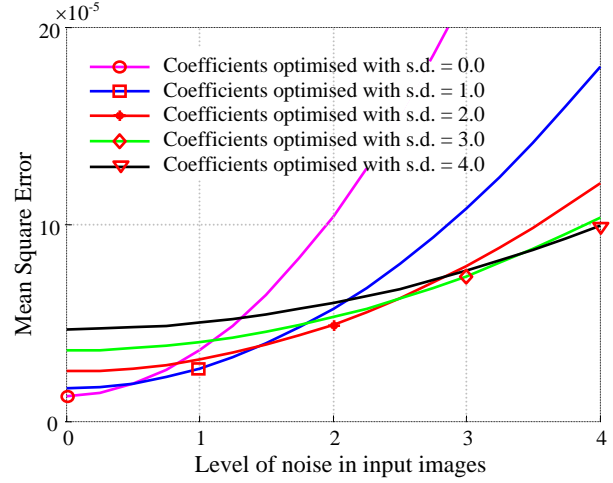


Figure 7: Performance of optimal interpolation on image 'beach' in the presence of noise using coefficients optimised at various levels of noise.

Figure 7 shows that for each input noise level, the best performance is achieved if the coefficients are optimised on the same level of noise. Now the same test can be applied to optimised coefficients on a different image. The procedure is exactly the same, only image 'sleep' is used to optimise the coefficients.

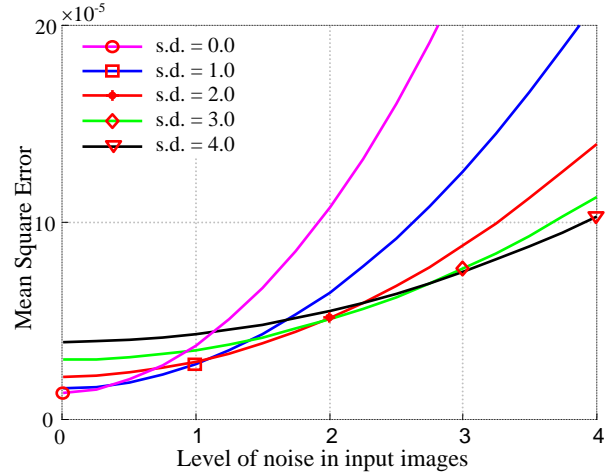


Figure 8: Performance of optimal interpolation on image 'beach' in the presence of noise using coefficients optimised on image 'sleep' at various levels of noise.

Figure 8 shows a similar trend with the coefficients being optimised on a different image. Hence, if the level of the noise in the input images can be estimated [10,11], the same amount (in the case of using image 'sleep' it can be seen that a slightly higher noise level is required) of noise can be used to optimise the coefficients. It can also be noticed that using coefficients estimated on noise levels within ± 1 of the input noise level yield satisfactory results, so the input level of noise needs to be estimated only approximately.

4 Conclusions

Non-uniform interpolation is a common procedure in image processing. This paper has focused on a new method of deriving the weights of a linear interpolation filter, which are optimal in a least squares sense. Based on the observation that the weights of a general linear interpolation filter depend only on the relative positions of the available samples, it was hypothesised that the optimal weights derived on one image would be near optimal on other images.

Experimentation showed that the optimal weights, derived through minimising the squared error between a known high-resolution image and a set of synthetically-created low-resolution images, are relatively independent of image content. Hence, weights optimised on a known image, can be used to interpolate an unknown image with the samples positioned in the same place as the known image.

While, in general, the desired high resolution pixel values are not available to calculate the optimal weights, this opens the possibility for near optimal interpolation using a synthetic image to derive the coefficients.

It was shown that in the presence of noise, the coefficients optimised at the same noise level as the input are likely to yield better results.

A future direction of work is to find an analytical way of deriving near optimal coefficients, hence decreasing the overhead. More experiments are planned to be completed to check the method on other test images and to use a variety of synthetic images to generate the weights.

5 References

- [1] S.C. Park, M.K. Park, and M.G. Kang, "Super-resolution image reconstruction: a technical overview," *IEEE Signal Processing Magazine*, vol. 20, pp. 21-36, May 2003.
- [2] A. Papoulis, "Generalized sampling expansion," *IEEE Transactions on Circuits and Systems*, vol. 24, pp. 652-654, 1977.
- [3] J. Yen, "On nonuniform sampling of bandwidth-limited signals," *IRE Transactions on Circuit Theory*, vol. 3, pp. 251-257, 1956.
- [4] N. Nguyen, P. Milanfar, and G. Golub, "A computationally efficient superresolution image reconstruction algorithm," *IEEE Transactions on Image Processing*, vol. 10, pp. 573-583, Apr 2001.
- [5] M. Kunter, K. Jangheon, and T. Sikora, "Super-resolution mosaicing using embedded hybrid recursive flow-based segmentation," in *The Fifth International Conference on Information, Communications and Signal Processing*, 2005, pp. 1297-1301.
- [6] D. Shepard, "A two-dimensional interpolation function for irregularly-spaced data," in *Proceedings of the 1968 23rd ACM national conference*, 1968, pp. 517-524.
- [7] S. Lertrattanapanich, and N. K. Bose, "High resolution image formation from low resolution frames using Delaunay triangulation", *IEEE Transactions on Image Processing*, vol. 11, pp. 1427-1441, Dec 2002.
- [8] D.G. Bailey, A. Gilman, and R. Browne, "Bias characteristics of bilinear interpolation based registration," in *IEEE Region 10 Conference (IEEE Tencon, Melbourne, Australia, (21-24 November, 2005)*, 2005.
- [9] G.R. Shorack, *Probability for Statisticians*, New York: Springer, 2000.
- [10] K. Rank, M. Lendl, and R. Unbehauen, "Estimation of image noise variance," *IEE Proceedings – Vision, Image and Signal Processing*, vol. 146, pp. 80-84, 1999.
- [11] G.E. Healey, and R. Kondepud, "Radiometric CCD camera calibration and noise estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 267-276, 1994.

Storing and Accessing Large Images using Summed Area Tables.

Volker Hilsenstein

Preventative Health Research Flagship, CSIRO Mathematical and Information Sciences
306 Carmody Road, St Lucia, QLD 4067, Australia.
Email: volker.hilsenstein@csiro.au

Abstract

This paper presents a simple, easy-to-implement method for storing and accessing large images with sizes of the order of gigapixels. The paper discusses the requirements for working with large images and examines how these are met in common image file formats. We propose a storage format, based on representing images as summed area tables, that addresses these requirements. Following a brief review of summed area tables we show how they can be used for fast downscaling of images. The storage costs of summed area tables are discussed. We report results obtained with a practical system to analyse large mosaics in automated microscopy and provide some benchmarks. The appendix gives practical tips on how to avoid restrictions in the standard C libraries of 32-bit operating systems, which can limit the size of files to 2^{31} bytes.

Keywords: summed area tables, integral images, mosaicing, image representation

1 Introduction: Working with Large Images

Large digital images with sizes on the order of gigapixels are becoming more and more common in fields such as biological and geospatial imaging. Such large images are typically mosaics created from a number of smaller images that are tiled or stitched together.

Creating very large images and accessing them in an efficient manner poses a data handling challenge because the images are typically too large to be held in memory on a standard PC. Although many of the commonly used image formats theoretically do not impose restrictions on image size, users typically run into practical limitations when the decompressed image is larger than the available RAM of their workstation.

In this paper we present a method for storing and accessing very large images based on summed area tables [1]. The technique is easy to implement and addresses some of the problems associated with large images such as efficient access to subregions and fast down-sampling.

Our adoption of summed area tables for storing large images was driven by a project described in a companion paper [2] in these proceedings. For that project, we acquired large image mosaics of histology sections using an automated microscope. The images were then analysed at different scales to extract and segment objects of interest. Af-

ter the analysis, much of the image data could be discarded or converted to compressed formats for archiving and therefore the increased storage requirements associated with summed area tables were not an issue (discussed in section 4.3.1).

The paper is organised as follows: Section 2 illustrates some of the challenges that arise when dealing with very large images, and reviews how these are addressed in common file formats. Section 3 reviews the definition of summed area tables and shows how they can be used in fast calculations of rectangle sums and thus for resampling of images. Section 4 presents our approach to working with and assembling summed areas on disk. In section 5 we present some results we obtained using our method for a typical application in microscopy. We conclude with a discussion in section 6.

2 Practical Size Limitations of Common Image Formats

Although most image file formats do not place restrictions on the image size, their practical use becomes limited for very large files. The limitations fall into two major categories:

a) Implementation Issues:

Prior to reading images from disk, many commonly available image libraries try to reserve a memory buffer that is large enough to hold the whole image. If the image size is much

larger than the available RAM, these memory allocation requests will fail and the image will not be read. Another implementation issue arises from the use of signed 32-bit integer file pointers, which restrict the maximum file size to 2^{31} bytes. In the appendix (section 7) we give some practical hints on how to circumvent the latter problem on common operating systems.

b) Image Representation:

For images that are too large to be loaded from disk as a whole, it becomes important that subregions can be accessed efficiently and that downscaled versions can be created without having to read in each individual pixel. The choice of image representation (raw pixel data, Fourier components, wavelets etc.) therefore becomes important. We discuss the suitability of common image representations in the following section.

2.1 Image Representations

The image representations used in common graphics file formats can be coarsely grouped into the following categories:

- Pixel-based.

The simplest image representations are pixel-based, with the intensity (or palette entry) of each pixel stored individually in a flattened array. For colour images, the colour triplets are typically stored either interleaved or as separate image planes. These image representations are basically raw formats, with some meta data in the image header describing the image size, bits per pixel etc. As each pixel can be individually addressed, it is easy and efficient to access subregions of the image on disk. However, extraction of downscaled previews is costly, because a smoothing kernel needs to be applied before sub-sampling, requiring access to every pixel (sub-sampling without prior smoothing leads to aliasing). Common file formats that support pixel-based representations are BMP, PNG and TIF.

- Fourier or DCT-based.

Frequency-based representations which store the Fourier or discrete cosine transform (DCT) coefficients of an image naturally lend themselves to down-sampling, because only the low-frequency coefficients need to be accessed to create a low-resolution overview image. However, because the spatial context is lost, subregions cannot be accessed quickly.

The most common format employing DCT coefficients, JPEG, addresses this issue by subdividing the image into small blocks of 8 by 8 pixels and computing the DCT coefficients on these. The small and fixed size of these blocks limits scalability when it comes to gigapixel-sized images.

- Wavelet-based.

Wavelet representations provide a decomposition of the image into different scales while retaining some of the spatial context. Thus they allow for quick scaling and efficient access to subregions. Common image formats supporting wavelet representations are JPEG2000 and OpenEXR. Closely related to wavelet representations are pyramids, which contain pre-computed, low-pass filtered versions of an image.

All these image representations can be compressed without loss of quality using techniques such as entropy coding, dictionaries or run length encoding. However, the ability to address individual pixels or frequency coefficients directly in the compressed bit stream is lost, thus making access to subregions less efficient. The Fourier, DCT and wavelet-based representations also lend themselves to lossy compression by suppression of coefficients with low magnitude.

Of the image representations listed above, wavelet-based image formats appear to be the gold standard for working large files, because they allow fast access to subregions and efficient down-sampling. Moreover, they allow for lossy compression, which can be an important consideration when building image archives. However, the relative complexity of wavelet-based file formats such as JPEG2000, combined with the lack of freely available implementations that can deal with gigapixel-sized images, limits the widespread application of this method. There are commercial JPEG2000 (for example Kakadu [3], ER Mapper [4]) implementations available that support gigapixel images, but the pricing of these libraries can make their use uneconomic for small, one-off commercial projects.

3 Summed Area Tables

Summed area tables form an array representation that allows for constant-time computation of pixel sums within any rectangular sub-window of an image. While the computational technique is likely to have been known much earlier, Crow [1] was the first to apply the technique to texture filtering in computer graphics. For texture filtering, summed

area tables form an alternative to pyramid-based interpolation techniques such as MIP¹-mapping[5]. In the computer vision community, summed area tables are sometimes referred to as *integral images*, a term introduced by Viola and Jones [6].

The following definitions are specific to the case of 2D images but they can easily be generalised to the n-dimensional case.

3.1 Definition

For a given 2D image I , the corresponding summed area table is computed as follows:

$$S(k, l) = \sum_{i=1}^k \sum_{j=1}^l I(i, j), \quad (1)$$

where $I(i, j)$ is the greyvalue at pixel position (i, j) . From equation (1) it is obvious that the summed area table is simply the (discrete) integral of the greyvalues.

The complete summed area table for an image can easily be computed using two passes, by first calculating the cumulative sums along all rows and then calculating the cumulative sums along all columns (or vice versa). This operation can be performed in-place, that is without an additional buffer, and requires $2nm$ additions for an image with $n \times m$ pixels. This means that the computation of the table is very fast because the cost scales linearly with image size.

3.2 Fast Computation of Rectangle Sums

Given the summed area table S corresponding to an image I , the sum of pixels in any rectangular subregion can be computed using four table look-ups and four additions:

$$\sum_{i=a_0}^{a_1} \sum_{j=b_0}^{b_1} I(i, j) = S(a_1, b_1) - S(a_0 - 1, b_1) - S(a_1, b_0 - 1) + S(a_0 - 1, b_0 - 1) \quad (2)$$

This is illustrated in figure 1. The mean greyvalue within any rectangle of an image I can thus be quickly obtained from S using equation (2) and dividing by the area of the rectangle.

Note that by applying equation 2 to single-pixel rectangles the original image can be reconstructed from its summed area table.

¹MIP stands for *multum in parvo*, “many in a small space”.

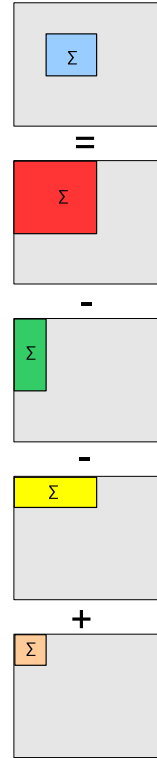


Figure 1: Computing rectangle sums using summed area tables.

3.3 Fast Resampling of Images

Because summed area tables can be used to calculate the mean greyvalue within any sub-window in constant time they lend themselves to application as low-pass filters. Thus, if one wishes to down-scale an image by a factor of n , one can divide the image into n by n -sized blocks, calculate the mean greyvalue for each block and then down-sample by n . This is equivalent to convolving the original image with an $n \times n$ -sized box filter before down-sampling.

Heckbert [7] presents an extension to the original idea that uses repeated integration of the original image and an increased (but still constant) number of table lookups to achieve filtering equivalent to smoothing with triangular, parabolic and higher-order kernels.

4 Using Summed Area Tables to Store and Access Large Images on Disk

Fast downscaling and efficient access to subregions were some of the requirements we identified for working with large images in section 2. The results from the previous section show that storing summed area tables on disk fulfils these requirements: it takes nm/k^2 hard disk read operations

to extract a k -times downsampled version from an $n \times m$ -pixel subregion, for arbitrarily large images².

4.1 Storage Layout

On the hard disk (as well as in memory), the 2-dimensional summed area table must be flattened into a 1D structure. For our implementation, we store the raw values of the summed area table as a serial vector in row-major order, with a short file header indicating the data type and the image dimensions.

4.2 Colour Images

Extending the storage layout presented in the previous section to colour images is straightforward, and can be achieved, for example, by interleaving the colour triplets for each pixel or by storing the three colour channels as separate planes.

4.3 Combining Multiple Summed Area Tables

Because the goal is to work with images that are too large to be held in RAM, the summed area tables have to be built up on disk. A simple method for this is to first write all the raw pixel values into the file at their respective array positions. Once all the pixels are written, the summed area table is then calculated in-place using two passes as described in section 3.1.

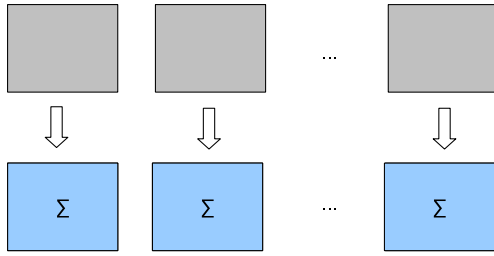
In this section, we present an alternative, tile-based approach for constructing the summed area table on disk. This approach was useful for our application in automated microscopy [2] where the tiles correspond to the individual images acquired by moving the microscope stage stepwise to positions on a regular grid. The tile-based construction is sketched in Figure 2 for combining tiles horizontally.

In a first step, image tiles that are small enough to fit into memory are converted to summed area tables and written onto disk. During a second step, the tiles are then read line-by-line. For each line, the rightmost pixel value of the first tile needs to be added to all the pixels of the next tile and so on. The combined line can then be written to disk.

We omit a detailed discussion of vertical tiling of the images. The same technique as for horizontal tiling can be applied, with slight implementation differences arising from the use of a row-major layout for storage.

²In practise, the performance assessment is more complicated because hard disk access is block-based and because the costs for seeking and reading differ.

Step 1: For each image tile, calculate SAT in RAM and write whole tile to disk.



Step 2: Open all SAT source tiles, read a single line from each file into RAM, calculate sum and write line to destination file.

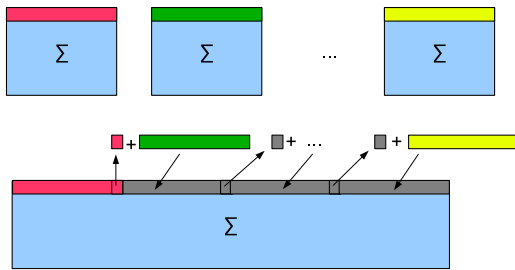


Figure 2: Combining summed area tables (denoted as SATs) horizontally tile-by-tile. It is assumed that the individual tables are stored in row-major order on disk.

4.3.1 Storage Cost

When working with summed area tables, care must be taken that the data type provides enough range to store the summed pixel intensities without overflow. For unsigned 8 bit integer images, a 10 gigapixel image can result in a maximum sum of $256 \cdot 10^{10}$. This requires 42 bits of storage, therefore a 32-bit integer type is not suitable for large tables. For gigapixel-sized image mosaics, the table values thus have to be stored as 64-bit integers or double-precision floating point values, leading to a substantial storage overhead.

5 Results

We devised a simple implementation of the proposed method in C. The implementation is naive in the sense that it does not include any optimizations, such as caching read and write operations or taking into account the block-based nature of file access. Table entries are represented as double precision floating point values in this implementation.

5.1 Benchmarks

We conducted a number of benchmarks on a standard PC (Pentium 4, 2.8 GHz, 1.5 GB Ram, 80 GB hard disk drive with 7200 rpm) running Windows XP. While conducting the benchmarks care was

taken that no other user applications placed a significant load on the system. However, system tasks may still have had some influence on the overall performance. Also, both the hard drive and the operating system cache data. In an attempt to reduce the influence of caching on the benchmarks, the different tests mentioned in sections 5.1.2 and 5.1.3 were interleaved, but caching may still have had some effect on the results.

5.1.1 Assembling a 1-Gigapixel Image

We recorded the time needed to assemble a gigapixel sized image from tiles using the method described in section 4.3. The summed area table was assembled from 32×32 image tiles containing 1024×1024 pixels each. The procedure was repeated 5 times. On average, the time required for creating the file was 52 minutes with a standard deviation of 4 minutes. Reading from the source tiles and writing to the final table was performed line-by-line and occurred on the same disk. The performance could have been improved by reading larger blocks and by using separate disks for reading and writing.

For comparison, we encoded the same gigapixel image as a JPEG2000 image using a lossless wavelet basis. The encoding was performed using a demonstration version of the ER Mapper JPEG2000 library [4] and took approximately six hours.

5.1.2 Downscaling

To assess the performance of downscaling using summed area tables, we recorded the time required for reading from disk and computing low-resolution versions of our 1-gigapixel image at different scaling factors. For each scaling factor, this was repeated 10 times. Figure 3 (top) shows the distribution of the times required for the downscaling as a function of the scaling factor. For each scaling factor, the boxplots summarise the median, the quartile ranges and the outliers of the time distribution. In the double-logarithmic plot, the median access times lie roughly on a line, as is to be expected based on the complexity of the downsampling operation discussed in section 4.

Note that the scaling factor shown for the horizontal axis of the plot refers to the scaling factor along each axis of the image. Thus, the area of the downsampled image decreases as the square of this factor.

5.1.3 Extracting subregions

To assess the speed at which subregions can be accessed, we extracted square regions of different

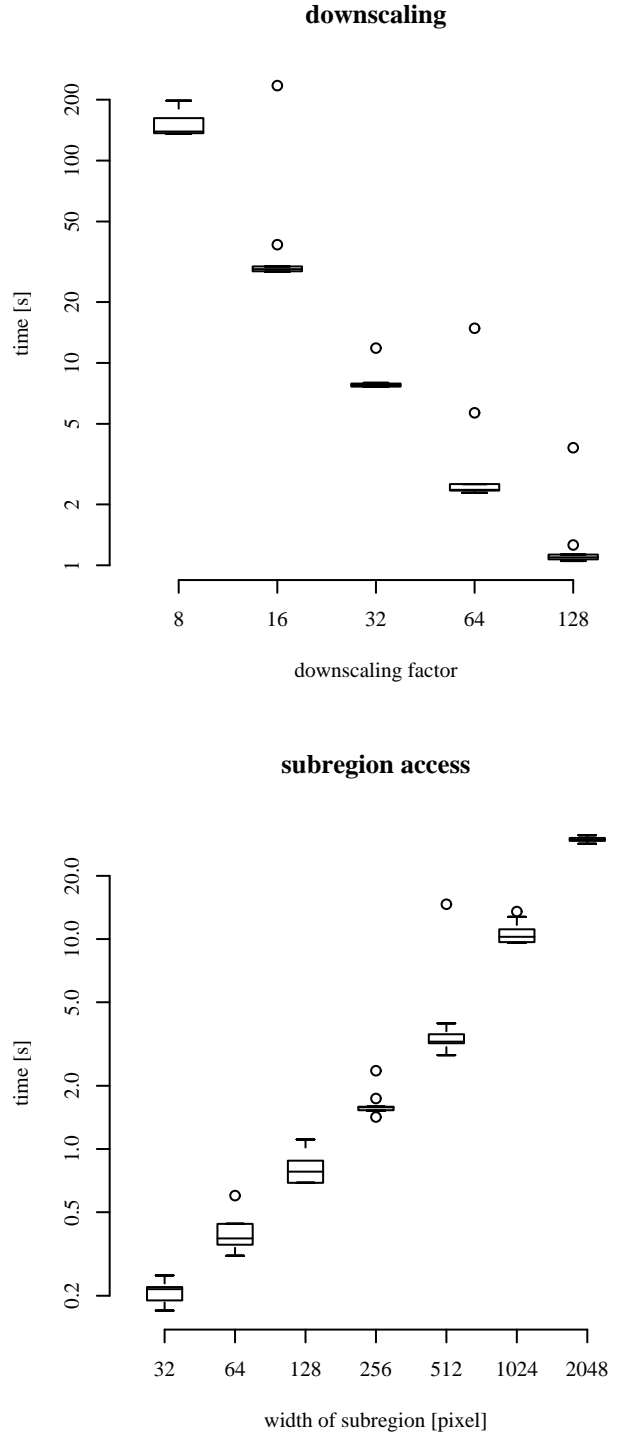


Figure 3: Benchmark results for downscaling of an image (top) and for accessing subregions (bottom) using summed area tables. See sections 5.1.2 and 5.1.3 for details.

sizes at full resolution from the summed area table. For each square size, this operation was repeated 10 times with subwindows located at different, randomly chosen positions within the summed area table. The recorded access times are shown

in Figure 3 (bottom) as a function of the width of the extracted squares. As for the downscaling benchmark, we use boxplots to summarize the time distributions.

5.2 Application Results

In addition to benchmarking, we have successfully applied the technique to a practical problem in automated microscopy, reported in our companion paper [2] in this issue. For that project, we scanned histology slides using an automated microscope. The resulting mosaic images were about 150 megapixels in size and were then analysed at different scales.

6 Conclusions

We have presented a method to work with gigapixel-sized images that is based on representing the images as summed area tables. We have applied the technique in multi-scale analysis of large mosaic images in microscopy.

The technique is attractive for small, one-off projects that require the handling of large images because it is very easy to implement and because many freely available image libraries are not suitable for the task. Technically, the method is not as advanced as wavelet-based methods, in particular the inflated storage requirements rule out the use in scenarios where collections of large images are to be archived. However, it can find its niche in scenarios as the one presented in [2], where much of the image data can be discarded or compressed after analysis. In particular, the conversion to summed area tables can be performed relatively quickly in comparison to encoding using wavelets.

Reducing the storage requirements of summed area tables could be a fruitful area for future work. For example, for many natural images, subtracting the mean intensity value from each pixel could prevent the area sums from accumulating to very high values, and would thus allow for storage using smaller data types. Also, a compression scheme that retains fast access to individual table entries could lead to a much wider applicability of the technique.

7 Appendix: Large Files on 32-bit Operating Systems.

Currently, all modern operating systems support file systems that can hold individual files which are larger than 2^{31} bytes. Nevertheless, when using the `fseek()` and `ftell()` functions in the standard C library one runs into problems when

trying to move the file pointer by more than 2^{31} bytes. These problems arise from the fact that the file offset pointer is stored as a signed 32 bit integer. To ensure correct operation with large files on Unix-based systems (including Mac OS X), one should resort to using the functions `fseeko()` and `ftello()` or `fseeko64()` and `ftello64()` instead. In addition, it may be necessary to define one or both of the preprocessor macros `_LARGEFILE_SOURCE` and `_FILE_OFFSET_BITS=64`. On 32-bit Windows systems `_lseeki64()` and `_telli64()` provide equivalent functionality.

References

- [1] F. C. Crow, "Summed-area tables for texture mapping," in *SIGGRAPH '84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, (New York, NY, USA), pp. 207–212, ACM Press, 1984.
- [2] V. Hilsenstein, P. Jackway, and P. Allingham, "An automated system for microscopy imaging and analysis of histology sections with an application in sheep meat morphometry," in *Proceedings Image and Vision Computing New Zealand 2006*, this issue, November 2006.
- [3] Kakadu Software, "Jpeg2000 software toolkit version 5.2.2." <http://www.kakadusoftware.com/>, visited on 10/9/2006.
- [4] ER Mapper, "ECW Jpeg2000 SDK 3.1." <http://www.ermapper.com/ecw/>, visited on 10/9/2006.
- [5] L. Williams, "Pyramidal parametrics," in *SIGGRAPH '83: Proceedings of the 10th annual conference on Computer graphics and interactive techniques*, (New York, NY, USA), pp. 1–11, ACM Press, 1983.
- [6] P. Viola and M. Jones, "Robust real-time object detection.," in *IEEE Int. Workshop on Statistical and Computational Theories of Vision - Modeling, Learning, Computing, and Sampling*, (Vancouver, Canada), July 2001.
- [7] P. S. Heckbert, "Filtering by repeated integration," in *SIGGRAPH '86: Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, (New York, NY, USA), pp. 315–321, ACM Press, 1986.

A comparison of noise in CCD and CMOS image sensors.

K. Irie¹, A.E McKinnon², K. Unsworth², and I.M. Woodhead¹

¹Lincoln Ventures Ltd, P.O. Box 133, Lincoln, Christchurch 7640, New Zealand.

²Applied Computing Group, Lincoln University, Canterbury, New Zealand.

Email: iriek@lvl.co.nz

Abstract

CCD and CMOS image sensors are commonly used in industrial image processing applications. They are often described using measures such as frame-rate, resolution, dynamic range, sensitivity, and read noise. However, noise detail within a captured image has a significant impact on subsequent image processing and is much more complex than read noise alone. This paper presents a model of the combined noise sources present in current digital video cameras, and compares measured noise between a commercially available CCD video camera and a CMOS video camera.

Keywords: CCD image sensor, CMOS image sensor, image analysis, noise measurement.

1 Introduction

Advancements in digital image sensors have led to the wide use of digital cameras in image processing applications. Many of these applications attempt to extract useful information from the images, which is fundamentally limited by the images' signal-to-noise ratio (SNR). The acquisition and conversion of photons in CCD and CMOS sensors is well documented in the literature and there are many references to the sources of image sensor noise [1-4]. However, there is little work on the comparison of the noise characteristics of CCD and CMOS image sensors. This study aims to model and compare the prominent noise sources in these sensors by statistical measurement, providing an objective method to compare their noise performances.

2 CCD/CMOS architecture

An overview of CCD and CMOS image sensors pertinent to this study is given. Photons are captured and converted to charge in the photo detection process. The charge is amplified, sampled, and digitally enhanced for output. Figure 1 shows a typical digital image sensor for a CCD/CMOS digital camera system [3].

Light passing through the sensor optics falls onto the imaging sensor. Many image sensors use microlenses to increase the amount of light incident on the photodetectors [1, 3, 5]. This also helps to reduce the problem of vignetting where, due to the optical tunnel formed by the sensor manufacturing process, light entering the sensor at an angle that is not parallel to the optical axis is attenuated prior to reaching the photodetector. In many colour cameras the light passes through a colour filter array (CFA) in order to generate trichromatic images. The filtered light then

enters the photodetectors where, depending upon the sensor fabrication method and design, approximately half of the incident photons are converted to charge [6].

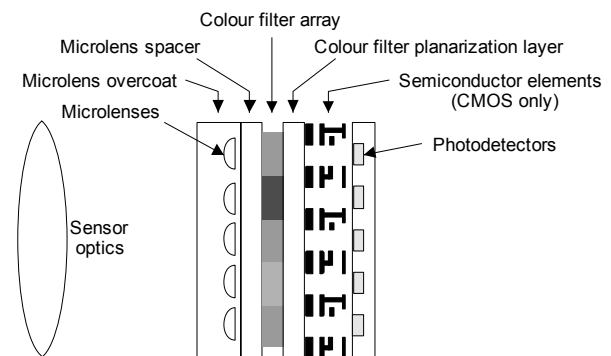


Figure 1: The components of a typical digital image sensor.

2.1 CCD image sensors

There are several readout architectures used for CCD sensors such as frame-transfer (FT), interline transfer (IL), and frame-interline transfer. IL CCD is the most popular image sensor for camcorders and digital still cameras, but suffers from a reduced fill rate (photon capture area) due to charge storage buffers located beside each pixel [1, 7].

This reduction in fill rate varies, but fill rates between 20-50% are not uncommon. The reduction is often compensated by using a microlens filter to increase the effective area of incident light collection. Figure 2 shows a typical IL CCD readout architecture. The charge is read sequentially, with each charge moving along a column or row in a conveyor type fashion. In CCD image sensors the analogue-to-digital conversion (ADC), storage, and enhancement are performed on supporting integrated circuits (IC's).

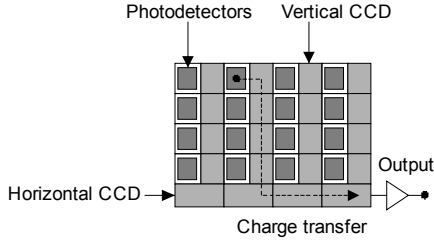


Figure 2: Architecture of an interline-type CCD sensor.

2.2 CMOS image sensors

Current CMOS sensors can be divided into two main architectures – passive-pixel sensors (PPS) and active-pixel sensors (APS) [1, 3, 8]. Figure 3 shows a typical readout architecture for a CMOS PPS. Each pixel contains a photodiode and one MOS transistor. As in most CMOS image sensors, row and/or column decoders are used for addressing the pixels in the array. Although they have relatively high fill rates due to having only a single transistor, PPS devices suffer from high noise due to large capacitive bus loads during pixel readout. They are also prone to column fixed-pattern-noise (FPN) from variations in the manufacturing process of the column amplifiers, which can result in objectionable vertical stripes in the recorded image.

CMOS APS can be divided into three main subtypes: photodiode, photogate, and pinned photodiode, where in each type three or more transistors are used in each pixel. APS typically have a fill rate of 50-70%, but the reduced capacitance in each pixel due to the transistor amplifiers leads to lower readout noise, which increases the signal-to-noise ratio (SNR) and sensor dynamic range (DR). The pinned-photodiode APS has currently been reported as the most popular implementation for CMOS image sensors [1].

There are other types of APS available but they are currently not in widespread use. The logarithmic photodiode sensor [8] operates continuously and

provides increased dynamic range from logarithmic encoding of the photocurrent. However, low output swing during low illumination and significant temperature dependence on the output are serious drawbacks limiting the use of this method. Fowler et al. [9] describe a ‘digital-pixel sensor’ that has 22 transistors and ADC functionality at each pixel.

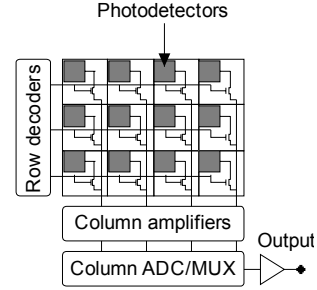


Figure 3: Architecture of a passive-pixel CMOS image sensor.

3 Camera noise model

The camera noise model is shown in figure 4¹. This model allows for the measurement of camera noise from the analysis of output images alone, which is necessary when evaluating an entire camera system where there is no access to the internal components of the camera. A description of each noise source is given in tables 1 and 2.

The equation for CCD image noise capture, N_{CCD} (from figure 4) is:

$$N_{CCD} = (I \times PRNU + SN_{ph}(I) + I + PFPN + SN_{dark} + N_{read}) \times N_D \times N_{filt} + N_Q, \quad (1)$$

where I is the sensor irradiance. The equation for CMOS image noise capture, N_{CMOS} (from figure 4) is:

$$N_{CMOS} = (((I \times PRNU + SN_{ph}(I) + I + PFPN + SN_{dark} + N_{read}) \times AFPN_{gain} + AFPN_{off}) \times CFPN_{gain} + CFPN_{off}) \times N_D \times N_{filt} + N_Q, \quad (2)$$

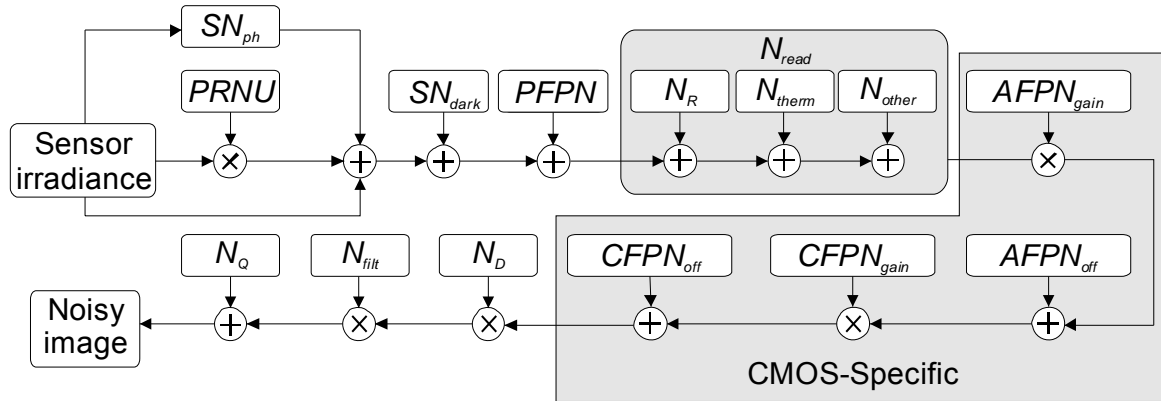


Figure 4: The noise model for image capture in a standard CCD/CMOS camera.

¹ Currently unpublished. Details available from authors.

Table 1: Description of modelled CCD camera noise sources.

Noise type	Symbol	Manifestation	Description	Dependencies
Passive offset fixed-pattern noise.	$PFPN$	Spatial variance.	Spatial offset of pixel values due to device mismatches during fabrication and their associated dark currents. Alternatively known as dark-signal non-uniformity (DSNU).	Temperature, exposure time.
Photo response non-uniformity.	$PRNU$	Spatial variance.	The difference in pixel responses (gain) to uniform light sources, due to differences in pixel geometry, substrate material, and microlenses.	Incident pixel illumination.
Photon shot noise.	SN_{ph}	Temporal variance.	Arises from random fluctuations in sampling of photons (Poisson noise).	Incident pixel illumination.
Readout noise.	N_{read}	Temporal variance.	A combination of noise sources attributed to the reading of pixel information: reset noise, thermal noise (Johnson-Nyquist), flicker noise, transistor dark currents, and other minor contributors.	Temperature, CCD readout rate.
Dark current shot noise.	SN_{dark}	Temporal variance.	Arises from random fluctuations in the number of dark current electrons (Poisson noise).	Temperature, exposure time.
Demosaicing effect.	N_D	Noise amplification or attenuation.	Arises from the interpolation of the colour-filter array to generate RGB triplets for each pixel.	Demosaicing implementation, combined sensor noise.
Digital filter effects.	N_{filt}	Noise amplification or attenuation.	Arises from digital effects like gain, contrast, and gamma.	Camera parameters.
Quantization noise.	N_Q	Additive noise, image content dependent.	Truncation or rounding of signals adds noise that becomes prominent when there is little variation in the image compared to the quantization step.	Variance of image data. Sets lower noise limit for non-trivial image content.

Table 2: Description of additional modelled CMOS camera noise sources.

Noise type	Symbol	Manifestation	Description	Dependencies
Active offset fixed-pattern noise.	AFP_{off}	Spatial variance.	The active elements in each CMOS pixel have variations in offset that contribute to image noise.	Temperature.
Active gain fixed-pattern noise.	AFP_{gain}	Spatial variance.	The active elements in each CMOS pixel have variations in gain that contribute to image noise.	Predominantly temperature and incident pixel illumination.
Column offset fixed-pattern noise.	$CFPN_{off}$	Spatial (across columns) variance.	Most CMOS image sensors incorporate column amplifiers to readout column data in parallel. Amplifiers each have an offset that vary across columns.	Temperature.
Column gain fixed-pattern noise.	$CFPN_{gain}$	Spatial (across columns) variance.	Gain variation in amplifiers described in $CFPN_{off}$.	Predominantly temperature and incident pixel illumination.

4 Method of Noise Measurement

The digital filtering in all tests was either disabled or set to neutral, effectively reducing N_{filt} to an identity function. The interpolated pixels are dependent upon the original pixels, and Bayer array demosaicing attenuation can be calculated by measuring the effect of the interpolated (averaged) pixels over the entire demosaiced image. Both calculated and measured results gave the following values for bilinear interpolation demosaicing attenuation:

$$\begin{aligned}
 \sigma_{demosaic,R} &= 0.73\sigma_R \\
 \sigma_{demosaic,G} &= 0.75\sigma_G \\
 \sigma_{demosaic,B} &= 0.73\sigma_B
 \end{aligned}
 \tag{3}$$

where σ_x is the standard deviation prior to demosaicing, and $\sigma_{demosaic,x}$ is the standard deviation of the demosaiced image. Quantization noise was shown to be dependent upon variation of image detail

and was measured to have a maximum value of $\sigma=0.29$, in accordance with quantization noise theory [10].

Video images of a GretagMacbeth ColorChecker Color Rendition Chart² (GMB color chart) were taken in an environment with controlled fluorescent and incandescent lighting³ and the chart was positioned such that it filled as much of the image frame as possible. The camera was deliberately set out of focus to reduce the effect of high-frequency content in the observed image that could affect the noise analysis. The illumination sources were positioned above the camera and directed towards the chart such that the image was free from direct specular reflection. For analysis of CCD camera noise a *Unibrain Fire i400 IEEE1394* camera was used, and for CMOS camera noise a *uEye UI-1210-C USB* camera was used. Both use Bayer CFAs, with images captured in their native resolutions of 640x480 pixels and 8-bits resolution

² <http://www.gretagmacbeth.com>

³ Lighting was empirically chosen such that the cameras' RGB responses to grey-scale colours were similar.

per R, G, and B channel. An image from the *uEye* camera is shown in figure 5.

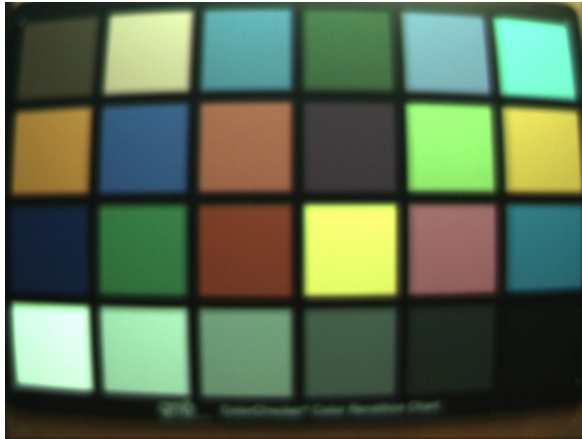


Figure 5: An example image of the GMB colour chart using the CMOS *uEye* camera.

Images of grey-scale panels on the 4th row of the GMB color chart (panel numbers 19-24) were analysed. These represent a grey scale from ‘white’ (Munsell value 9.5) to ‘black’ (Munsell value 2). An area of 50 pixels by 50 pixels was extracted from the centre of the image of each panel for analysis. This ensured that the area of analysis was located within each panel so that edge effects from panel boundaries were avoided.

Two calibration procedures were executed prior to measurement of noise content to provide a uniform basis for measurement:

- (i) Stuck or ‘hot’ pixels were identified by capture of an image with long exposure in dark conditions. The stuck pixels were removed from processing.
- (ii) The camera was tested for sensor homogeneity. Images were taken with no illumination and total image noise was measured across the GMB panels 19-24.

Environmental and sensor temperatures were monitored and non-temperature related experiments performed at an ambient temperature of approximately 22°C.

The standard deviation (σ) was chosen as the statistical measure for the analysis of image noise as it can be easily related to the magnitude of pixel variations. The mean intensity of each pane μ_p , taken as the mean of the extracted 50x50 panel, was used as a measure of I . For this study, 100 images achieve a 95% confidence interval for the resulting image analysis.

Three methods for noise analysis were formulated: analysis of temporal noise, analysis of spatial (fixed-pattern) noise, and analysis of the total combined spatial and temporally varying noise for each row and column of image data. These three methods enabled

the measurement of all noise components for constant exposure and temperature:

(i) Temporal noise:

σ of temporal data, σ_t , for each pixel was calculated over the set of 100 images.

σ_t for all pixels were then averaged, giving a value $\mu(\sigma_t)$ for the mean temporal variation for the panel.

(ii) Spatial (fixed-pattern) noise:

1. The mean of the temporal data, μ_t , for each pixel was calculated over the set of 100 images.

2. A second-order polynomial fit for each column of μ_t was calculated and subtracted from the data to remove optical effects such as vignetting and $1/R^2$ illumination fall-off expected from the use of a discrete illumination source.

The residuals after subtraction of the polynomial-fitted data were concatenated and σ calculated to determine a value of $\sigma(\mu_t)$ for the mean spatial variation for the panel.

(iii) Total image noise:

1. A second-order polynomial fit was calculated for each data row and column of an image of a GMB panel. The fitted line was subtracted from the data to remove optical effects such as vignetting and $1/R^2$ illumination falloff expected from the use of a discrete illumination source.

2. The residuals after subtraction of each fitted line were concatenated and σ of the concatenated data calculated to give row and column noise values σ_r and σ_c respectively.

3. σ_r and σ_c for the panel were averaged over 100 test images to derive the final noise figures $\mu(\sigma_r)$ and $\mu(\sigma_c)$ for the panel for each image set.

N_Q and N_D were measured on simulated images with various levels of added Gaussian noise.

5 Results

The noise quantities in tables 1 and 2 were measured and are listed in table 3. Figure 6 shows the total measured and modelled noise response for the *i400* CCD camera. All 3 channels exhibit the same trend, although at different offsets. The overall noise results illustrate a reasonable fit between the measured and modelled data. Only measured data values below 140 were used as the camera response started saturating above this value. Figure 7 shows the modelled noise response for the *uEye CMOS* camera. The response between colour channels is consistent, although it has a different curve than that measured with the CCD camera. Overall the CCD camera exhibits more noise than the CMOS camera.

Table 3: Measured CCD and CMOS camera noise sources at 22°C ambient temperature.

Noise quantity	Measured CCD (σ)	Measured CMOS (σ)
$PFPN$	R=0.30 G=0.12 B=0.25	R=0.14 G=0.10 B=0.05
$PRNU$	R=0.010 μ_R -0.008 G=0.006 μ_G +0.122 B=0.013 μ_B +0.024	R=max(0, 0.0038 μ_R -0.57) G=max(0, 0.0038 μ_G -0.53) B=max(0, 0.0038 μ_B -0.63).
SN_{ph}	R=0.21 $\sqrt{\mu_R}$ + 0.468 G=0.52 $\sqrt{\mu_G}$ + 0.01 B=0.22 $\sqrt{\mu_B}$ - 0.04	R=0.10 $\sqrt{\mu_R}$ - 0.15 G=0.09 $\sqrt{\mu_G}$ + 0.09 B=0.10 $\sqrt{\mu_B}$ - 0.08
N_{read}	R=1.61 G=0.61 B=1.24	R=0.60 G=0.58 B=0.60
SN_{dark}	=0	=0
N_D	R=0.73R G=0.75G B=0.73B	R=0.73R G=0.75G B=0.73B
N_Q	≤ 0.29	≤ 0.29
AFP_{off}	N/A	R=1.00 G=1.00 B=1.15
AFP_{gain}	N/A	R=1.003 μ_R G=1.003 μ_G B=1.002 μ_B
CFP_{off}	N/A	R=0.119 G=0.091 B=0.034
CFP_{gain}	N/A	R=1.016 μ_R G=1.019 μ_G B=1.008 μ_B

Figure 8 shows the relative magnitudes of the noise components for the blue channel of the *i400* CCD camera. Quantization noise only becomes significant with noise variations less than $\sigma=1$, and so is not a major contributor to image noise in this camera. In low illumination conditions N_{read} dominates but becomes less significant with increasing illumination, as $PRNU$ and SN_{ph} increase significantly. $PFPN$ is a minor source of noise. The red channel response is similar to the blue channel's, but the green channel exhibits a greater contribution from SN_{ph} and lower contributions from $PFPN$, $PRNU$, and N_{read} .

Figure 9 shows the relative magnitudes of the noise components for the blue channel of the *uEYE* CMOS camera, which exhibits a more complex response due to the additional sources of noise. The reduction in $PFPN$ in the CMOS camera is more than offset by the effects of AFP_{off} , due to the active elements at each pixel (note the scale differences between figures 8 and 9). Measured $PRNU$ is significantly reduced in the CMOS camera, which is one of the major differences between the overall noise responses of the two

cameras. In low illumination the noise in the *uEye* camera is dominated by N_{read} and AFP_{off} . SN_{ph} becomes significant at higher illumination levels, with increased AFP_{gain} and $PRNU$ as well. All colour channels exhibit similar noise-component responses. N_Q sets the lower limit of measurable noise in real-world digital images.

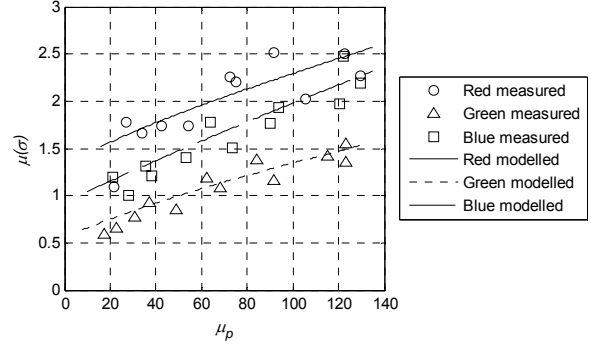


Figure 6: The total measured and modelled noise for the *i400* CCD camera.

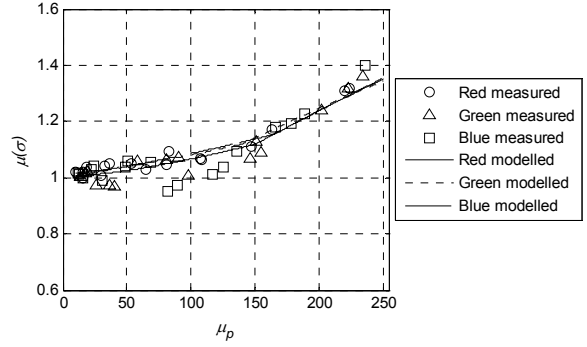


Figure 7: The total measured and modelled noise for the *uEYE* CMOS camera.

6 Discussion

The differences in response to photon shot noise, SN_{ph} , between the cameras illustrate the maximum number of photons that each camera's image sensor can capture. A camera will scale its dynamic range to fit within a particular number range (0-255 in the two cameras tested), and so a lower photon shot noise coefficient results in a greater number of photons captured. Hence the CMOS camera (1/3 inch sensor) which has a shot noise coefficient of approximately 0.1 for all colour channels is able to capture more photons than the CCD camera (1/4 inch sensor⁴) which has a noise coefficient of greater than 0.2.

The CCD camera exhibits variations between colour channels, which may in part be due to the CCD camera having some internal colour balancing to take into account the design of the CCD.

⁴ The actual sensor size is less than the quoted 'value', as the historic measurement of sensor size is based on an analogue tube size, not the size of the image capturing element.

Of particular interest is the *PRNU* response of the CMOS camera. *PRNU* only starts adding to the image noise at incident illumination greater than approximately 50% of the dynamic range of the sensor. The effect of *CFPN* is reasonably minor, but has a visual impact on the final image as the eye is sensitive to constant spatial variations (like vertical stripes from *CFPN_{off}*) that make the column noise appear worse than it actually is.

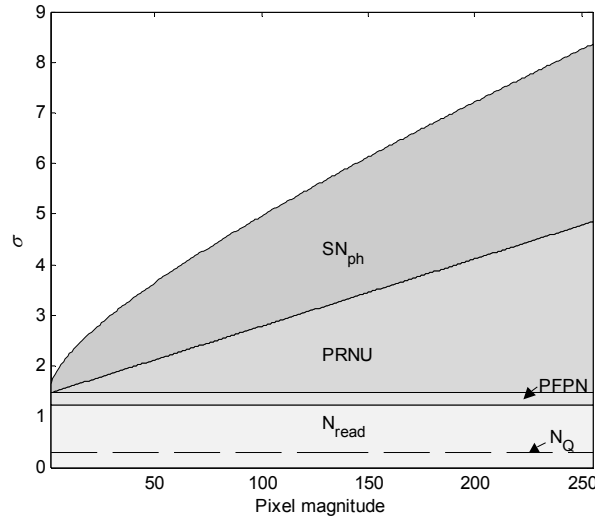


Figure 8: The relative measured magnitudes of CCD noise components in the blue channel of the *i400* camera.

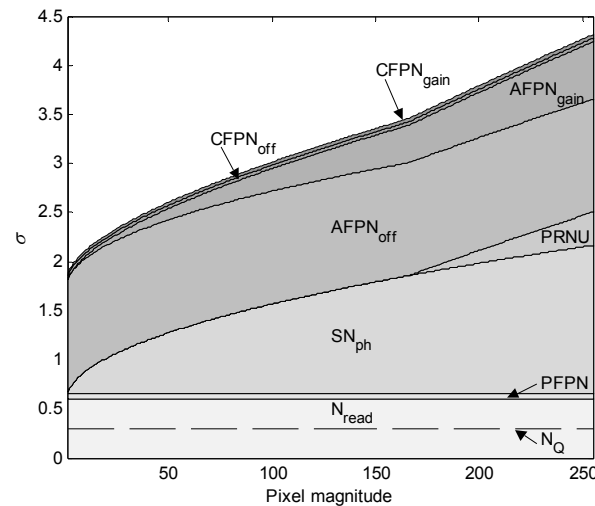


Figure 9: The relative measured magnitudes of CMOS noise components in the blue channel of the *uEYE* camera.

The overall noise quantities for both cameras have a roughly equal contribution from spatial and temporal noise sources. It may be possible to reduce spatial noises in software by subtraction of a calibrated spatial noise image, although care would be required as many sources of spatial noise are temperature dependent. Apart from quantization noise, the noise level under illumination is bound by photon shot noise.

Selection of an appropriate sensor for industrial image processing depends on a variety of factors. In the case of the *i400* and *uEye* cameras tested, the *uEye* exhibits higher consistency between colour channels and exhibits less noise than the *i400* camera.

7 Acknowledgements

This work was supported by the New Zealand Foundation for Research, Science and Technology programme LVLX0401.

8 References

- [1] J. Nakamura, *Image Sensors and Signal Processing for Digital Still Cameras*: CRC Press, 2006.
- [2] R. E. Flory, "Image acquisition technology," *Proceedings of the IEEE*, vol. 73, pp. 613-637, 1985.
- [3] A. El Gamal and H. Eltoukhy, "CMOS image sensors," *Circuits and Devices Magazine, IEEE*, vol. 21, pp. 6-20, 2005.
- [4] R. Costantini and S. Susstrunk, "Virtual Sensor Design," *Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications V, Proceedings of SPIE*, vol. 5301, pp. 408-419, 2004.
- [5] T. Chen, P. B. Catrysse, A. El Gamal, and B. A. Wandell, "How small should pixel size be?," *Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications, Proceedings of SPIE*, vol. 3965, pp. 451-459, 2000.
- [6] L. Brouk and Y. Nemirovsky, "CMOS SOI image sensor," *International Conference on Electronics, Circuits and Systems, Proceedings of the IEEE*, vol. 11, pp. 156-159, 2004.
- [7] "Kodak CCD Primer #KCP-001," Eastman Kodak Company - Microelectronics Technology Division 1999.
- [8] O. Yadid-Pecht and A. Fish, "Active Pixel Sensor Design - from pixels to systems," in *CMOS Imagers: From Photo-transduction to Image Processing*, Yadid-Pecht and Etienne-Cummings, Eds.: Kluwer, 2004.
- [9] B. Fowler, A. El Gamal, and D. X. D. Yang, "A CMOS area image sensor with pixel-level A/D conversion," presented at 1994 IEEE International Solid-State Circuits Conference, 1994. Digest of Technical Papers. 41st ISSCC, 1994.
- [10] H. Baher, *Analog and digital signal processing*: John Wiley & Sons Ltd., 1990.

Pros and Cons of the Nonlinear LUX Color Transform for Wireless Transmission with Motion JPEG2000

T. Totozafiny¹, F. Luthon¹, and O. Patrouix²

¹Computer Science Lab LIUPPA, University of Pau, IUT Chateau Neuf 64100 Bayonne, France.

²Laboratory for Industrial Process and Services ESTIA, Technopole Izarbel 64210 Bidart, France.

Email: t.totozafiny@estia.fr; Franck.Luthon@univ-pau.fr; o.patrouix@estia.fr

Abstract

We present in this paper an investigation of the nonlinear *LUX* color transform in a system based on Motion JPEG2000 intended for road surveillance application. The system allows capturing an image, performing motion detection, encoding with the JPEG2000 coder, and transmitting the image towards the decoder through the GSM network at a rate of one image per second. The decoder reconstructs and displays the received image. The *LUX* color transform is used instead of standard linear color transforms like *YUV*. The results show that the color rendering of the reconstructed image is clearly improved.

Keywords: JPEG2000 coding, ROI, Reference image, Logarithmic color transform, Wireless transmission

1 Introduction

JPEG2000 is an ISO/ITU-T still image coding standard developed by the Joint Photographic Experts Group (JPEG). JPEG2000 is designed to provide numerous capabilities within a unified system. One interesting option of JPEG2000 is the Region Of Interest coding (ROI), when certain parts of the image are of higher importance than the background. In JPEG2000 Part 1 [1], the core coding algorithm, two linear color transforms are proposed as standard: reversible or irreversible, resulting in lossless or loosely coding respectively. The Logarithmic hUe Extension (LUX) is a non linear color transform, referring to the biological human vision system. It can improve the JPEG2000 coding results [3]. A smart encoding based on JPEG2000 coder was developed in [4]. Each frame is independently coded using the ROI option. Then, data are transmitted toward a decoder where a final image is reconstructed. The transmission is made with two layers. The first layer contains the data, the second layer contains the reference image. But the authors do not explain how to obtain the initial reference. The typical problem of this kind of system is the updating of the Remote Reference Image (RRI). Recently [7], we have developed a new system allowing image transmission through a single channel with very low bandwidth (GSM). The transmission rate reaches the performance of one image per second. The updating of the RRI is triggered when certain conditions are met (linked to the amount of motion areas such as: too many, few or no moving objects). Instead of standard linear color transforms, for example the *RGB* to *YCrCb* transform, the LUX transform is used in order to improve the color rendering. Here, we investigate its advantages and drawbacks in our system [7]. This paper is organized as follows: we describe our system in Section 2 and the presentation of the non linear LUX color transform is given in Section 3. Finally, experimental results, discussion and conclusion are given in Sections 4, 5, and 6.

2 Proposed System

2.1 System Description

The system is intended for the transmission of color image series through a wireless GSM network. The image is captured by a static camera. Then, it is JPEG2000-coded and transmitted toward a decoder where the final image is reconstructed and displayed (Fig. 1). First, the system builds a reference image and transmits it towards the decoder. This initialization is necessary in order to reconstruct the final image for the next frame transmission. After the initialization is completed, the reference image is regularly updated. Motion detection is performed in order to obtain an ROI mask, which gives the region of interest for the system. Then, the current image containing only data linked with the mask is coded using the ROI

option of JPEG2000 standard and transmitted towards the decoder. At the decoder side, the image is received and the motion mask is implicitly reconstructed. The final image is built using the available reference image, the motion image and the reconstructed mask. In the spatial domain, a simple pixel substitution is used to build the displayed image. The RRI must be updated according to the scene evolution. Hence, a flag is added in the file header of the bitstream during the image encoding in order to tag the sent image, which can be a reference image (i.e. background image) or a motion image (i.e. motion data). In the first case, the received image is stored as background image on the decoder. Otherwise the image is displayed.

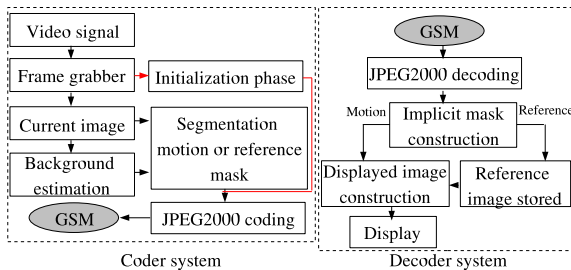


Figure 1: Our coder and decoder system.

2.2 Initialization step

The initialization phase consists in the construction of a reference image. It is a difficult task because it depends not only on the moving objects present in the scene (such as object entering, stopping at and/or leaving the scene, object moving slowly, etc) but also on the natural environment (sun, movements due to wind, illumination, shadows and so on). In a static camera framework, several methods in the computer vision literature were proposed to obtain a reference image during an initialization phase [8]. The proposed system is intended for videosurveillance. The coder system is an embedded architecture working on a PC104 format board where the memory size is restricted. In fact, we use the first order recursive filter :

$$B_{init}(p, t + 1) = \alpha_p I(p, t + 1) + (1 - \alpha_p) B_{init}(p, t) \quad (1)$$

where $B_{init}(p, t)$ and $I(p, t)$ are the intensity values at the pixel location p at time t in the reference image and the current image respectively. $\alpha_p \in [0, 1]$ is the learning rate. This model (1) gives good results only when the moving objects are small. Otherwise, the quality of the regions covered by a large object is bad in the reference image obtained. Therefore, it will be useful to improve these regions first.

2.3 Foreground extraction

The motion detection is a binary labelling problem [6]. It consists in putting, on each pixel p of image I at time t , one of the two following label values:

$$e_p = \begin{cases} 1 & \text{if } p \in \text{moving object} \\ 0 & \text{if } p \in \text{static background} \end{cases} \quad (2)$$

In order to carry out the binary labelling we use two observations. For each pixel p , we compute:

- the difference between the reference image and the current image:

$$o_{dr}(p, t) = |B(p, t) - I(p, t)| \quad (3)$$

- the difference between two successive frames:

$$o_{dt}(p, t) = |I(p, t) - I(p, t - 1)| \quad (4)$$

To find the most probable label with these two observations, we can use the conditional probability (Bayes theorem). In our application, we used a logical AND in order to cope with the embedded architecture target. We put a threshold θ on both observations o_{dr} and o_{dt} and then we compute the logical AND. The moving object threshold θ is determined automatically, according to an entropy based threshold selection. A morphological gradient filter is applied to improve the binary mask obtained. Then, the current image is compressed using the ROI option with a very low bit rate (9600 bps for single GSM channel) which yields a transmission rate of 1 img/s.

2.4 Reference image updating at ground station

We use the same scheme as developed in [6] to estimate a reference image in the embedded device. For each grabbed frame the background image is updated. The method consists in the modelling of each pixel temporal evolution with K Gaussian distributions ($K = 3$).

2.5 Reference image updating at the remote station

Once the initialization phase is completed, the coder is able to send the entire background image. Thus, the decoder can reconstruct the final image.

At the coder side, the reference image is updated for each frame according to a Gaussian Mixture Model (GMM). The updating of the reference image at the decoder (RRI) should be done regularly.

The basic technique for updating the RRI is based on the transmission of the RRI towards the decoder within a specific period, for example every 4 seconds or every 10 images like in [4]. The RRI must be coded with no ROI option and with a moderate bit rate compression. In our system, the transmission of the complete RRI would slow down the overall transmission rate. For our application, image transmission should be done at a rate of 1 img/s at least, while using a single transmission channel. The complete RRI update takes more than 4 seconds. This latency is not acceptable. In order to keep the image transmission rate close to 1 img/s, we propose a new updating technique of the RRI by pieces. In this scheme, the RRI will be coded like the motion image with a ROI mask. We have chosen a square pattern in order to build the ROI mask. The information in this area will be coded and sent to the decoder for the updating of the background image (Fig. 2).

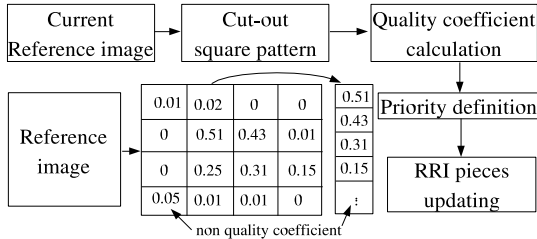


Figure 2: RRI updating strategy.

2.5.1 Regions to be updated

The background image is composed of Nb blocks. Each block will be used as an ROI mask. In order to improve the efficiency of this strategy, we define a parameter \overline{qc} for each area. \overline{qc} represents a non quality coefficient corresponding to the initialization phase (background image creation) or the GMM background estimation. \overline{qc}_i ($0 \leq \overline{qc}_i \leq 1$) is used to compute a refreshing priority for region i . The region with the highest priority will be updated first.

2.5.2 Local quality coefficient computation

In a given frame, the local quality coefficient for a given block is based on the intersection of the motion ROI mask and the block area. The occupation percentage of the moving object is computed as follows:

Let $Imask$, Ir be the mask of moving object and current reference image respectively, and i be the current index of the treated block.

We can write:

$$Ir = \bigcup_{i=0}^{Nb-1} Irb_i \quad Imask = \bigcup_{i=0}^{Nb-1} Imaskb_i \quad (5)$$

where Irb , $Imaskb$ are sub-images of Ir and $Imask$ respectively. For each block i , we compute:

$$\overline{qc}_i = \frac{NonZero(Imaskb_i)}{Nsize^2} \quad (6)$$

where $NonZero()$ is an operator that counts the number of pixels set to 1 in the current $Imaskb$ and $Nsize$ is the size of the square block.

2.5.3 Global quality coefficient computation

At frame t , the quality coefficients are updated according to the local quality coefficients and the previous values of the quality coefficient at frame $t-1$. A threshold θ_{qc} is introduced to determine a changing of given region:

$$\overline{qc}_i(t) = \begin{cases} \max(\overline{qc}_i(t-1), \overline{qc}_i(t)) & \text{if } \overline{qc}_i(t) \geq \theta_{qc} \\ (1-\gamma)\overline{qc}_i(t-1) & \text{else} \end{cases} \quad (7)$$

where γ is the learning rate for the background estimation. During the initialization phase for example, with model of Eq.(1), we have : $\gamma = \alpha_p$.

2.5.4 Remote reference updating decision

The region to be updated in remote reference must correspond to the block with the highest \overline{qc} . We introduce two thresholding values θ_{high} and θ_{down} in order to measure the amount of place that moving objects take in the scene. The high threshold θ_{high} indicates that a lot of moving objects are observed in the scene while θ_{down} indicates that few moving objects are observed in the scene. The strategy of RRI updating is engaged according to the state (amount) of moving objects in the scene (many or few). Three cases are considered:

- case 1: $\theta_{state} < \theta_{down}$,
- case 2: $\theta_{state} > \theta_{high}$,
- case 3: $\theta_{down} < \theta_{state} < \theta_{high}$.

where θ_{state} is the global state of moving objects altogether. It can be computed as:

$$\theta_{state} = \frac{NonZero(Imask)}{dim} \quad (8)$$

where $Imask$ is a mask for the moving object and dim represents the image size.

2.5.5 Blocks pulling

The blocks to be updated are chosen according to the three cases previously stated.

- case 1: we can consider that no interesting object is present in the scene. We propose to use the block with the highest \overline{qc} as the ROI mask for the JPEG2000 encoding. Then, the non-quality coefficient of this block is forced to zero ($\overline{qc} = 0$) in order to flag it as “treated”.
- case 2: too many moving objects are detected in the scene, then the updating of the remote reference should not be engaged (only motion is transmitted). But if this configuration lasts too long (for example more than 2 hours), maybe the updating should be performed like in the previous case.
- case 3: there is no easy decision so we choose to force the updating every n frames (typ. value of $n = 20$) in order to improve the global quality of the displayed image. Only the blocks with a \overline{qc} greater than θ_{qc} will be updated. Then, the non-quality coefficient of this block is forced to zero ($\overline{qc} = 0$) in order to flag it as “treated”.

3 Logarithmic color transform

3.1 LUX color space introduction

In the JPEG2000 coding part one, the multicomponent color transform is carried out using a linear color transform, either $RGB \rightarrow YUV$ or $RGB \rightarrow YCrCb$, allowing a reversible coding as well as an irreversible coding respectively. The conversion matrices, forward and inverse, of the irreversible coding are:

$$T = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.16875 & -0.33126 & 0.5 \\ 0.5 & -0.41869 & -0.08131 \end{bmatrix}$$

$$A = T^{-1} = \begin{bmatrix} 1.0 & 0 & 1.402 \\ 1.0 & -0.34413 & -0.71414 \\ 1.0 & 1.772 & 0 \end{bmatrix}$$

The nonlinear LUX color transform (for Logarithmic hUe eXtension) is inspired by the biological human vision system (Fig. 3). The human eye is a natural compression system and the compression is done nonlinearly: the cone transduction function may be described by a loglike function, while the action of horizontal and bipolar cells (weighted average and weighted difference resp.) may be modelled by a linear matrix like T . The logarithmic image processing (LIP) model is known to yield an impressive contrast enhancement [5].

This one LIP is basically defined in the continuous case by three equations: a transform f from the intensity space (variable x) to the space of tones, an isomorphism ϕ from the space of tones into a logarithmic space (variable y) and an inverse isomorphism ϕ^{-1} (for more details, see [2]).

The *LUX* color space extends the LIP model to handle colors (i.e., $YCrCb$). For that purpose, only the composition function $\Phi = \phi \circ f$ is of practical interest. The isomorphism Φ provides a logarithmic transform normalized by the maximum transmitted light:

$$\begin{aligned} \Phi : x &\rightarrow y = M \ln \frac{x_0}{x} \\ \Phi^{-1} : y &\rightarrow x = x_0 \exp -\frac{y}{M} \end{aligned}$$

where $x \in]0 \dots x_0]$ is a continuous gray level, $x_0 \in]0 \dots M]$ is the maximum transmitted light and M is the dynamic range of gray levels (typ. $M = 256$ for 8-bit coding).

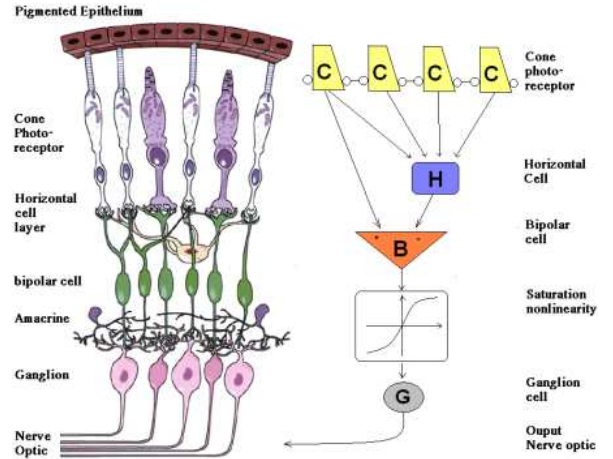


Figure 3: Biological analogy.

From a mathematical point of view, the diagram below helps understand how the *LUX* is built by composition of functions:

$$(R, G, B) \xrightarrow{Norm} (r, g, b) \xrightarrow{\Psi} (l, u, x) \xrightarrow{Denorm} (L, U, X) \quad (9)$$

3.2 LUX forward transform

First, color components are normalized. To adapt the dynamic range, the normalization consists in two steps: translation of dynamics and rescaling of the quantities w.r.t. their maximum values. Since $(R, G, B) \in [0, M] \times [0, M] \times [0, M]$ in the discrete case, translated quantities (r, g, b) are defined to stick to the interval $]0, M]$ as required by the LIP theory, yielding normalized quantities $\overline{R}, \overline{G}, \overline{B}$. Let (r_0, g_0, b_0) be the maximum values of (r, g, b) . We have:

$$r = R + 1 \quad g = G + 1 \quad b = B + 1$$

$$\bar{R} = \frac{r}{r_0} \quad \bar{G} = \frac{g}{g_0} \quad \bar{B} = \frac{b}{b_0}$$

Then the nonlinear transform Ψ which is the composition of three functions $\Phi^{-1} \circ T \circ \Phi$ may be computed as:

$$l = \bar{R}^{t_{11}} \bar{G}^{t_{12}} \bar{B}^{t_{13}}$$

$$u = \bar{R}^{t_{21}} \bar{G}^{t_{22}} \bar{B}^{t_{23}}$$

$$x = \bar{R}^{t_{31}} \bar{G}^{t_{32}} \bar{B}^{t_{33}}$$

where t_{ij} are coefficients of matrix T .

So far, this logarithmic model works only for positive values of chrominances. To take account of the possibly negative values of the chromatic components, we can use the following restriction:

$$\bar{u} = \begin{cases} \frac{u}{2} & \text{if } u \leq 1 \\ 1 - \frac{1}{2u} & \text{if } u > 1 \end{cases}$$

$$\bar{x} = \begin{cases} \frac{x}{2} & \text{if } x \leq 1 \\ 1 - \frac{1}{2x} & \text{if } x > 1 \end{cases}$$

Finally, a proper denormalization step yields the three nonlinear color components $\in [0, 255]$:

$$L = Ml - 1$$

$$U = M\bar{u} - 1 \quad (10)$$

$$X = M\bar{x} - 1$$

To simply the Eq.(10), we use the following expression to compute the LUX forward transform:

$$L = (R + 1)^{t_{11}} (G + 1)^{t_{12}} (B + 1)^{t_{13}} - 1$$

$$U = (R + 1)^{t_{21}} (G + 1)^{t_{22}} (B + 1)^{t_{23}} - 1$$

$$X = (R + 1)^{t_{31}} (G + 1)^{t_{32}} (B + 1)^{t_{33}} - 1 \quad (11)$$

3.3 LUX inverse transform

The inverse of Eq.(11) is :

$$R = (L + 1)^{a_{11}} (U + 1)^{a_{12}} (X + 1)^{a_{13}} - 1$$

$$G = (L + 1)^{a_{21}} (U + 1)^{a_{22}} (X + 1)^{a_{23}} - 1 \quad (12)$$

$$B = (L + 1)^{a_{31}} (U + 1)^{a_{32}} (X + 1)^{a_{33}} - 1$$

where a_{ij} are coefficients of inverse matrix $A = T^{-1}$. In our experiment, we use (11) and (12) to perform the forward and inverse LUX color transform respectively.

4 Experimental results

The system has been implemented into the embedded device (working with a PC104 format board). This one is mainly based on a camera, a CPU and a wireless transmission device. The system can currently send an image at a rate of one image per second through the RS232 port restricted

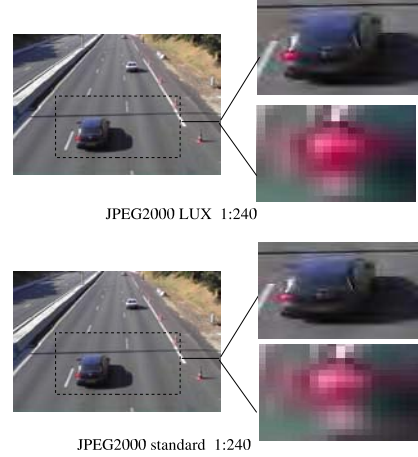


Figure 4: A particular region coded with JPEG2000 ROI option at bit rate 0.1bpp ($1 : 240$): nonlinear color transform $RGB \rightarrow LUX$ (top) ; standard color transform $RGB \rightarrow YCrCb$ (bottom).



(a) (b)

Figure 5: Reconstructed images: a) standard color transform ($RGB \rightarrow YCrCb$); b) LUX color transform.

to 9600 bps. For experimental tests we used 2 PCs. For JPEG2000 encoding we used the Kakadu SDK (<http://www.kakadusoftware.com>). Then we changed the standard linear color transform by LUX . We can see on Fig. 4 that LUX gives a better color rendering (rear light on left) than the standard color transform (here we have used the irreversible one). Fig. 5 shows a result of the reconstructed image at the decoder side. The size of the image is 320×240 with 8 bit-coding per sample. It is coded on a low bit rate compression 0.1bpp . The PSNR is used to evaluate the result of reconstructed image using the standard color and LUX . The PSNR computation for LAPS¹ video is given on Fig. 6. We can see that the LUX color transform result is better than the standard color transform. We gain between 1 and 5dB, but the average computation time is five times more, as shown Tab. 1.

¹Thanks to LAPS laboratory <http://www.u-bordeaux1.fr> for providing a sequences images; We have also tested our algorithm with the sequences available at http://i21www.ira.uka.de/image_sequences/

Table 1: Average computation time (in *ms*) under Linux system.

Color Transform methods	PC AMD 1.4GHz	Embedded PC104 Celeron 800GHz
Standard	180	250
<i>LUX</i>	600	900

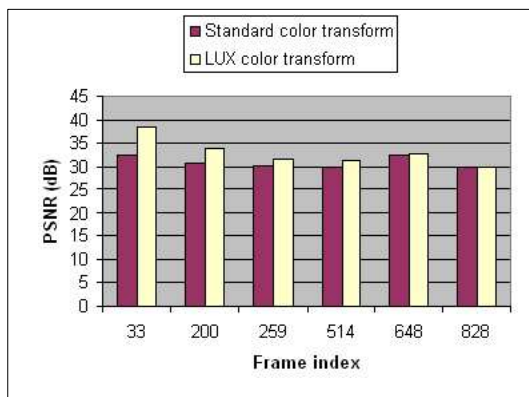


Figure 6: PSNR computation result of LAPS video, compressed and transmitting image at 1 : 240 ratio compression and *1img/s* respectively

5 Discussion

We note that *LUX* is not a standard transform of the JPEG2000 still image coding scheme. The user that hopes to use an available hardware JPEG2000 coder on the market is wedged because the *LUX* color transform is not implemented. To accomplish the inverse color transform, in general the maximal values of the colors dynamic range will be sent to the decoder separately from the codestream coded by JPEG2000. Then the computation time of *LUX* is very expensive (see Tab. 1). Nevertheless, the logarithmic hue extension *LUX* improves the rendering quality of the image in high compression applications, as shown with PSNR computation (Fig. 6). In road surveillance applications, obtaining the best color restitution is always an advantage. For example when one car is overtaking another one, capturing the flashlights is essential. Then, finding a good compromise between the *LUX* computation time and the color rendering may be interesting. As the power of processors increases (Moore law), the computation time of the nonlinear color transform *LUX* could be considerably decreased in the future. In the perspective, the fast implementation of *LUX* computation by means of either software or hardware will be interesting. We would use JPEG2000 non compliant in this case.

6 Conclusion

The *LUX* computation time is expensive and it is not implemented in the JPEG2000 coding standard. Yet, in a road surveillance context we have developed a system allowing an image transmission at a rate of one image par second through the GSM network. The image is coded on a low bit rate compression $0.1\text{bpp}(1 : 240)$. The rendering of color can be improved using nonlinear color transforms like the logarithmic hue extension color transform.

References

- [1] ISO/IEC JTC 1/SC 29/WG 1 (ITU-T SG8). JPEG2000 Part I Final Committee Draft Version 1.0. Technical report, Mars 2000.
- [2] M. Jourlin and J. Pinoli. Image dynamic range enhancement and stabilization in the context of the logarithmic image processing model. *Signal Processing*, (41):225–237, 1995.
- [3] F. Luthon and B. Beaumesnil. Color and R.O.I. with JPEG2000 for wireless videosurveillance. In *IEEE Int. Conf. on Image Processing, ICIP'04*, Singapore, October 2004.
- [4] J. Meessen, C. Parisot, C. Lebarz, D. Nicholson, and J. F. Delaigle. Smart Encoding for Wireless Video Surveillance. In *SPIE Proc. Image and Video Communications and Processing*, San Jose, CA, January 2005.
- [5] S. Shah and M. D. Levin. Visual Information Processing in Primate Cone Pathways. *Part I A Model. IEEE Trans. on System, Man and Cybernetics*, (26):259–274, 1996.
- [6] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, pages 246–252, 1999.
- [7] T. Totozafiny, O. Patrouix, F. Luthon, and J. M. Coutellier. Dynamic Background Segmentation for Remote Reference Image Updating within Motion Detection JPEG2000. In *International Symposium on Industrial Electronics (ISIE2006)*, 9-13 July, ETS-Downtown Montréal, Canada 2006.
- [8] H. Wang and D. Suter. A Novel Robust Statistical Method for Background Initialization and Visual Surveillance. In *ACCV*, pages 328–337, 2006.

Moment-based Local Descriptor using Scale Invariant Keypoints

Jae-Sun Han, Gwang-Gook Lee, Whoi-Yul Kim

Division of Electrical and Computer Engineering, Hanyang University

Email: jshan@vision.hanyang.ac.kr

Abstract

Local descriptors are being widely used in many applications such as object recognition and image registration. Among many local descriptors, SIFT (Scale Invariant Feature Transformation) is one of the most popular descriptors due to its robustness under various image transformations. For better matching performance, improved local descriptors that combined the SIFT and other description methods have been recently proposed, for example PCA-SIFT and GLOH.

In this paper, two moment-based local descriptors are proposed and their performances are compared. The keypoints extracted by SIFT are combined with each moment based image descriptor. Zernike and ART are adopted to generate descriptions for keypoint regions. Both are appropriate for describing the contents of blob-like regions defined by SIFT keypoints since both have a set of basis functions that are defined in polar coordinates. Through experiments that took into account scale, rotation, and viewpoint changes, it was proven that the proposed local descriptors outperform the original SIFT algorithm.

Keywords: local descriptor, ART, Zernike

1 Introduction

Due to their invariance under geometrical transforms and robustness to occlusions, local descriptors are vigorously employed in diverse areas such as pattern recognition, machine vision, and computer graphics. For example, they play important roles in solving problems such as object recognition [1], image retrieval [2], and the creation of panoramas [3]. For these reasons, there have been many studies on local descriptors [4]. In a comparative study conducted by Mikolajczyk and Schmid, SIFT was proven to be the most stable local descriptor under geometrical transformations [6][5][2].

SIFT's calculation consists of two parts: 1) localizing keypoints invariant under geometric transforms and 2) extracting descriptor for selected keypoint regions [7]. For better matching performance of SIFT, there were several studies that replaced only the descriptor part while the original keypoint obtained using the SIFT algorithm was kept intact.

A method that employs, instead of the gradient-based descriptor of the original SIFT, PCA combined with the SIFT keypoints was proposed, and named PCA-SIFT. Compared with the original SIFT, PCA-SIFT showed better performance under geometrical image transforms including scale and rotation.

A similar method, named GLOH has been suggested as an alternative. GLOH also uses the SIFT keypoints, but employs generated descriptors in a different way [9]. GLOH computes gradient description consisting of three histogram bins in the radial direction and eight in the angular direction, whereas the original SIFT uses four bins for each of the x and y axis. GLOH showed better matching performance than the original SIFT under various image transformations.

In this paper, we propose new local descriptors that combine SIFT keypoints with moment-based image descriptors. Zernike moments [13] and ART (Angular Radial Transform) coefficients [10] are utilized as the image descriptors. Both Zernike and ART are computed in polar coordinate systems; hence it is natural to use these moments to describe the blob-like area defined by SIFT keypoints.

This paper is organized as follows. Chapter 2 provides a brief explanation of each of algorithm employed in this paper. Chapter 3 describes how SIFT keypoints, Zernike moments and ART are utilized for local description. Chapter 4 compares the performance of the proposed method with the previous methods (SIFT and PCA-SIFT) for real images. The experimental conditions include image scaling, rotation and view point changes via affine transforms. Finally, chapter 5 concludes this paper.

2 Background

In this section, a brief review of the algorithms adopted for the proposed method is provided. That is, SIFT keypoints and two moment-based descriptors, Zernike and ART.

2.1 SIFT

The SIFT, which was proposed by Lowe [7] combines a scale invariant keypoints detector and a descriptor based on the gradient distribution in the regions defined by detected keypoints.

The calculation process of SIFT can be divided into two parts: 1) selecting geometrically invariant keypoints, 2) creating local description of neighbour pixels.

First, a scale space of the input image is built by successive Gaussian convolutions with different variances. Then, DoG (Difference of Gaussian) images that approximate the LoG (Laplacian of Gaussian) image are generated from these scale space images.

Equation (1) represents the Gaussian convolution used to build the scale space. The DoG images are obtained by subtracting two Gaussian images as shown in Equation (2).

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (1)$$

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \quad (2)$$

Finally, local extrema in the DoG image are determined as keypoint positions. Since the keypoints are selected as the extrema points in a scale space, the same positions can be localized under scale change and rotation of the images. The size of the Gaussian kernel of the local maximum is used as the scale of the keypoint. The scale value is used to define the same area around the keypoint undergoing scale change.

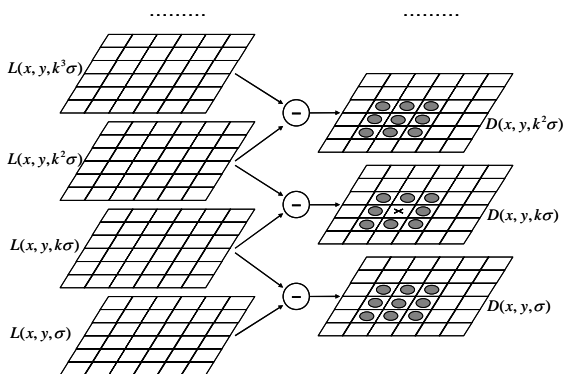


Figure 1. Calculation of DoG images and keypoints localization. Local extrema are selected as keypoints through comparison with 26 neighbours in 3x3 regions at the current and adjacent levels.

Figure 1 shows the process of keypoint localization. After DoG images are calculated in the scale space,

keypoints are selected only if they are larger than all of these neighbours or smaller than all of them.

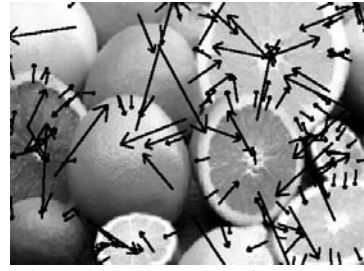


Figure 2. Selected local extrema in a DoG pyramid.

Figure 2 shows the SIFT keypoint vectors extracted from an image. In the figure, the origin of an arrow represents the position of a keypoint, and the direction of an arrow indicates the main orientation of the local area corresponding to the keypoint. The scale of the local area corresponding to each keypoint is expressed by the length of an arrow.

Each of the local areas is defined by keypoints in consideration of the variance of a DoG image where the keypoints are selected. In an area around a keypoint, the descriptor is created by sampling the magnitudes and orientation of the image gradient. It is represented by a 3D histogram of gradient locations (x and y direction) and orientations.

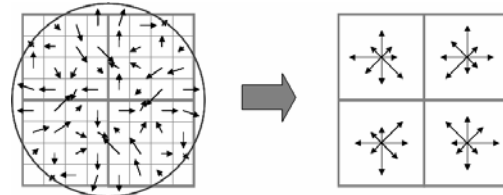


Figure 3. Descriptor creation process. The histogram of gradient consists of a total of 128 bins, which include 4 bins for each location in the x and y directions, and 8 bins for orientation.

The image on the left in Figure 3 is an example of extracted gradient values. The direction and magnitude of each of the gradients are expressed with an arrow. The image on the right shows orientation histograms summarizing the contents over 4x4 subregions of the image on the left, with the length of each arrow corresponding to the sum of the gradient magnitude near that direction within the region.

2.2 Zernike moments

Zernike polynomials are a set of complex polynomials that form a complete orthogonal set over the interior of the unit circle [13].

Its basis function $V_{nm}(x, y)$, of order n and repetition m is defined as follows.

$$V_{nm}(x, y) = V_{nm}(\rho, \theta) = R_{nm}(\rho) \exp(jm\theta) \quad (3)$$

In this formula, n is zero or a positive integer, and m is an integer that satisfies $n-|m|=(\text{even})$, and $|m| \leq n$. ρ is the distance from the origin to point (x, y) , and it is

valid over the range of $0 \leq \rho \leq 1$. Also, θ is the magnitude of the angle between point (x, y) and the x axis.

$R_{nm}(\rho)$, real polynomials of radial direction for the Zernike basis function, is defined as in equation (4).

$$R_{nm}(\rho) = \sum_{s=0}^{(n-|m|)/2} (-1)^s \frac{(n-s)!}{s! \left(\frac{n+|m|}{2} - s\right)! \left(\frac{n-|m|}{2} - s\right)!} \rho^{n-2s} \quad (4)$$

Figure 4 is a visualized example of the basis function of Zernike moments. It represents the real part of the basis function when m is even.

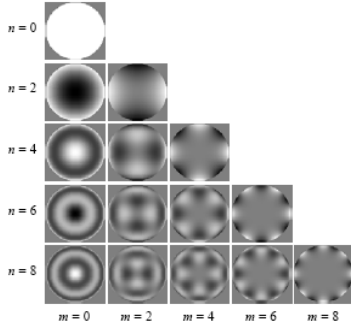


Figure 4. Real part image of Zernike basis function. The basis function of Zernike consists of complex polynomials. This figure visualizes the real part of the basis function.

Using this basis function, a Zernike moment about the image function $f(\rho, \theta)$, which is defined on the polar coordinate, is expressed as

$$Z_{nm} = \frac{n+1}{\pi} \iint_{\text{unit disk}} V_{nm}^*(\rho, \theta) f(\rho, \theta), \quad (5)$$

where V_{nm}^* is a complex conjugate of V_{nm} .

Each of calculated moments becomes a component of the moment vector. The similarity distance of between two Zernike moment vectors is calculated by summing up the weighted absolute differences of each moment, like the formulas in equation (6),

$$d_z(A, B) = \sum_{i=0}^{L-1} W_i \times \|M_A[i] - M_B[i]\|, \quad (6)$$

where L is the dimension of the Zernike moment vector, and $M[i]$ s are the i^{th} moment of each moment vectors, respectively. The weight, W_i can be the variance of each moment in the images, or it can be simply be set to 1.

2.3 ART

ART (Angular Radial Transform) is an orthogonal transform using a basis composed of sinusoidal functions on a polar coordinate system [10]. ART coefficients are calculated from the convolution between basis functions and image functions of a polar coordinate. A coefficient, of order n in the radial direction and m in the angular directions, can be expressed as follows.

$$F_{nm} = \langle V_{nm}(\rho, \theta), f(\rho, \theta) \rangle = \int_0^{2\pi} \int_0^1 V_{nm}^*(\rho, \theta) f(\rho, \theta) \rho d\rho d\theta, \quad (7)$$

where, $f(\rho, \theta)$ is the image function on the polar coordinate, and $V_{nm}(\rho, \theta)$ is the basis function of ART. $V_{nm}(\rho, \theta)$ can be divided into functions of radial direction $R_n(\rho)$ and angular direction $A_m(\theta)$, as in equation (8).

$$V_{nm}(\rho, \theta) = R_n(\rho) A_m(\theta)$$

$$R_n(\rho) = \begin{cases} 1, & \text{if } n=0 \\ 2\cos(\pi n \rho), & \text{otherwise} \end{cases} \quad (8)$$

$$A_m(\theta) = \frac{1}{2\pi} \exp(jm\theta)$$

Figure 5 shows a part of the basis functions as an image.

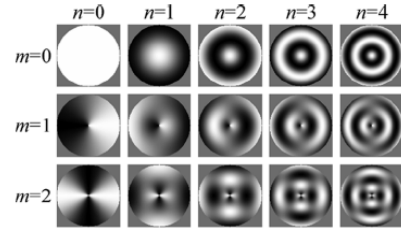


Figure 5. Image of real part of ART basis function.

This figure shows basis functions up to the 4th order in the radial direction and up to the 2nd order in the angular direction. The ART basis is similar to the Zernike basis in that it also consists of complex polynomials; however, Figure 5 shows an image only of the real part.

ART coefficients are used as descriptors after being normalized by $n=m=0$ coefficient components, and the similarity among descriptor vectors is determined by the distance between two vectors, $d_a(A, B)$.

$$d_a(A, B) = \frac{1}{NM-1} \sum_{i=0}^{NM-2} |ART_d^A[i] - ART_d^B[i]| \quad (9)$$

3 Proposed Local Descriptor

In this section of the paper, the proposed local descriptor, which combines SIFT keypoints and the moment-based descriptors, is described.

Since the SIFT algorithm selects keypoints from the local extrema of DoG (Difference-of-Gaussian) in a scale space of images, the keypoints can be extracted at the same point even when the image suffers geometric transforms such as scaling and rotation. Also, a keypoint keeps track of the information of its scale and the dominant orientation that are used to generate identical descriptions for the keypoint region.

As mentioned in the previous section, some of the previous studies employed only the SIFT keypoints and utilized other descriptors (rather than the original SIFT descriptor) to improve its matching performance. Similarly, the proposed method utilizes SIFT

keypoints with moment based image descriptors to compute local descriptors. Zernike and ART are selected as the moment based image descriptors.

Because keypoints extracted by SIFT are the local extrema of DoG, they are essentially blob-like features. Hence, both ART and Zernike moments, which have unit-circle bases defined on the polar coordinate system, are appropriate to describe the contents of the keypoint areas selected by the SIFT algorithm.

Moreover, Zernike moments and ART have the advantage of being invariant to rotation. The SIFT keypoints also contain the main direction of the local keypoint for dealing with rotations; however, the SIFT algorithm generates more than one descriptor for a single keypoint when there are a number of dominant orientations. This means that the SIFT can not completely cope with rotation changes, thus generating redundant descriptors.

Zernike moments and ART have two desirable properties as a descriptor for keypoints extracted by SIFT. One of these properties allows Zernike moments and ART to make up the weak points of the original SIFT, that is, multiple descriptors for one keypoint. In fact, it is desirable to generate a descriptor for a keypoint, if the descriptor is sufficiently invariant to rotation changes. Since Zernike and ART representation of a local area yields the similar description, that is, invariance to the orientation, it is possible to generate only one description for each keypoint. The other desirable property of Zernike moments and ART is their orthogonality. In general, moments are defined as the projection of a function $f(x, y)$ onto basis function set. Since both Zernike moments and ART have orthogonal basis sets, the contribution of each moment to the image becomes unique and independent.

ART is an improved version of Zernike that maintains the two desirable properties mentioned above. Since Zernike basis functions do not equally describe the radial and angular directional complexities, ART was developed from the motivation of making new basis functions to improve Zernike moments (Zernike basis functions are defined only when $m < n$). So it is not only rotation invariant and orthogonal, but also takes into account both description complexities in radial and angular directions.

Figure 6 represents the whole process of the proposed method. As illustrated in the figure, each of the local areas is represented by the extrema of DoG in the scale space, and descriptions are calculated by projecting local images to the basis functions of Zernike moments or ART.

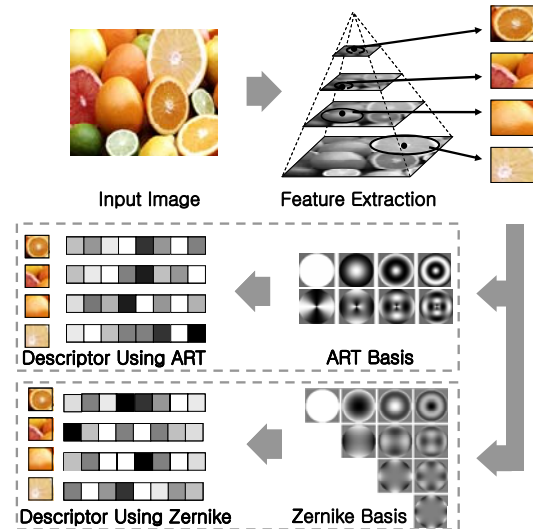


Figure 6. Extraction process of local descriptors.

4 Experimental Results

For the evaluation of the suggested two local descriptors, the matching performance under known artificial geometric changes was examined. The geometrical transform included rotation, scaling, and viewpoint change of images. We also used each of the 35 ART coefficients (radial 3, angular 12) and 35 Zernike moments (up to order 10) except the first component which was used for the normalization.

The experimental results were compared with those of the original SIFT and PCA-SIFT. The performance of each descriptor was presented with the ratio between positively matched correspondences and total features.

$$C = \frac{\text{Positive Match}}{\text{Total Features}} * 100 \quad (10)$$

Figure 7 shows samples of the test images. Rotation and scale changes were applied so that each image undergoes geometric distortions for that arbitrary viewpoint.

For rotation, we measure the accuracy by rotating the images from 10° to 50° . Table 1 lists the results of the experiment with rotation changes, and Figure 8 illustrates the results. According to the results, both methods of using Zernike moments and ART showed better performance than SIFT and PCA-SIFT. As expected, ART slightly outperformed Zernike moments.

Table 2 and Figure 9 show the experimental results for scale changes. SIFT and our method of using ART slightly outperformed Zernike. Notably however, all three greatly outperformed PCA-SIFT.

With regard to viewpoint and complex changes, proposed methods also outperformed SIFT and PCA-SIFT. The results are presented in Table 3 and Figure 10.

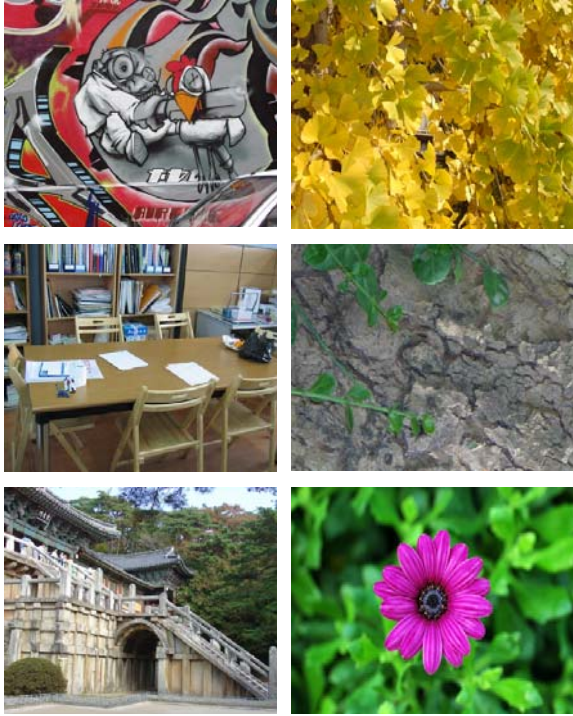


Figure 7. Test images.

Table 1: Rotation change experimental results.

	10	20	30	40	50
SIFT	71.63	70.58	70.53	70.88	70.15
PCA-SIFT	73.29	72.21	72.45	72.35	71.52
Zernike	75.72	74.62	74.69	75.24	74.65
ART	76.18	74.80	75.15	76.16	75.87

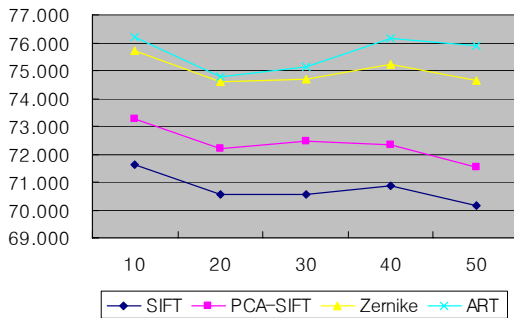


Figure 8. Matching performance of rotation changes.

Table 2: Scale change experimental results.

	0.9	0.8	0.7	0.6	0.5
SIFT	75.37	80.04	78.60	80.47	78.88
PCA-SIFT	73.37	76.68	75.33	76.85	75.93
Zernike	75.95	79.10	78.95	79.92	77.91
ART	76.05	79.76	79.53	80.73	78.56

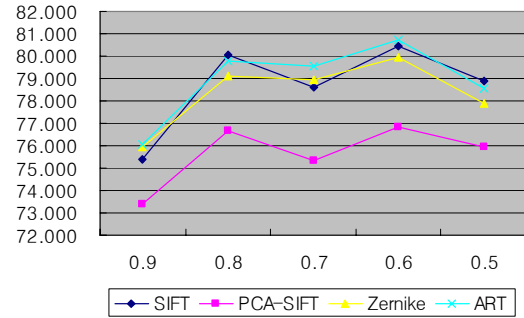


Figure 9. Matching performance of scale changes.

Table 3: Viewpoint and complex change experimental results.

	x=20, 0.7	x=30, 0.8	x=10, 0.6	x=40, 0.7	y=10	z=10
SIFT	59.63	59.85	61.24	59.01	69.23	70.96
PCA-SIFT	56.18	57.02	58.46	55.67	68.14	69.16
Zernike	61.79	62.03	63.77	61.06	70.15	71.37
ART	62.94	63.05	64.74	62.94	69.67	71.70

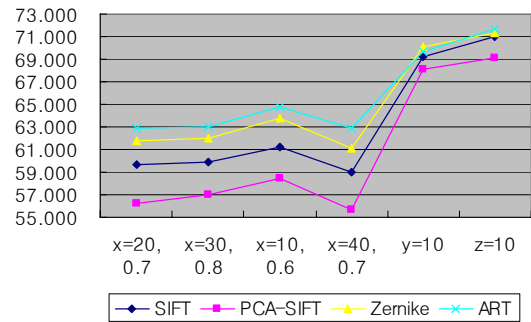


Figure 10. Matching performance of affine and complex changes.

5 Conclusion

New local descriptors that combine SIFT keypoints with either ART or Zernike moments are proposed in this paper. In experiment, both proposed methods performed better than the original SIFT or PCA-SIFT, considering rotation, scale, and view point changes. Especially for the experiments with rotation, new local descriptors notably outperformed previous methods due to their unique property, the rotation invariance.

Although the new descriptors showed better performance for geometrical changes, they suffer from large illumination changes. To cope with this problem, means of obtaining full illumination invariance are under investigation.

In future studies, the proposed descriptors will be applied to practical applications such as object recognition and image retrieval. In addition, other means of improving the proposed local descriptors, for example, ways to improve the extraction process

of keypoints in colour images, will be also investigated.

6 References

- [1] M. Brown and D. Lowe, "Invariant Features from Interest Point Group", In British Machine Vision Conference, BMVC, pp 656-665, 2002.
- [2] K. Mikolajczyk and C. Schmid. "Indexing based on scale invariant interest point", In Proceedings of International Conference on Computer Vision, pp 525-531, July 2001.
- [3] M. Brown and D. Lowe, "Recognising Panoramas", International Conference on Computer Vision (ICCV 2003), Nive, France, pp 1218-1225, Oct. 2003.
- [4] Florica Mindru, Tinne Tuytelaars, Luc Van Gool, and Theo Moons, "Moment invariants for recognition under changing viewpoint and illumination", Computer Vision and Image Understanding, vol.94, No.1-3, pp 3-27, Apr. 2004.
- [5] W.Freeman and E. Adelson, "The Design and Use of Steerable Filters," IEEE Tras, Pattern Analysis and Machine Intelligence, vol.13, no. 9, pp 891-906, Sept. 1991.
- [6] K. Mikolajczyk and C. Schmid, "A Performance Evaluation of Local Descriptors", In Proceedings of Computer Vision and Pattern Recognition, July 2003.
- [7] D. Lowe, "Distinctive image features from scale invariant keypoints", In International Journal of Computer Vision, vol 60, pp 91-100, 2004.
- [8] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local", Computer Vision and Pattern Recognition (CVPR), pp 511-517, 2004.
- [9] K. Mikolajczyk and C. Schmid, "A Performance Evaluation of Local Descriptors", Pattern Analysis and Machine Intelligence, IEEE Transactions, vol 27, No. 10, pp 1615-1630, Oct. 2005.
- [10] Whoi-Yul Kim and Yong-Sung Kim, "A new region-based shape descriptor: The ART(Angular Radial Transform) Descriptor", ISO/IEC/JTC1/SC29/WG11/MPEG99/M547 2, Maui, Dec. 1999.
- [11] M. Border, J. -D. Kim, Y.-S. Kim, W.-Y. Kim, K. Muller, "Summary of the results in shape descriptor core experiment", ISO/IEC JTC1/SC29/WG11/M4869, MPEG Meeting, Vancouver, Canada, July 1999.
- [12] S.-K. Hwang and W.-Y. Kim, "Fast and Efficient Method for Computing ART", Image Processing, IEEE Transaction, vol 15, NO. 1, pp 112-117 Jan. 2006.
- [13] Whoi-Yul Kim and Yong-Sung Kim, "A region-based shape descriptor using Zernike moments", Signal Processing: Image Communication 16, pp 95-102, 2000.

A Hybrid Approach to Man-Made Structure Extraction from Natural Scenes

Hang Zhou, David Suter, and Konrad Schindler

Institute for Vision Systems Engineering, Dept Elect. & Comp. Syst. Eng.
PO Box 35, Monash University, Clayton, VIC 3800, Australia
{hang.zhou, d.suter, konrad.schindler}@eng.monash.edu.au

Abstract

We present a new approach that integrates supervised segmentation formulated in a Bayesian network, and fractal feature based unsupervised segmentation. The result is a more reliable algorithm for man-made structure extraction from natural scenes. A causal Multi-Scale Random Field (MSRF) model is used as the prior model on the class labels of the image sites, and a Gaussian Mixture Model (GMM) connects the label field to the image data. Instead of solely relying on supervised training, a rough unsupervised division of the image, prior to the accurate segmentation is performed. In this pre-segmentation, multi-scale fractal dimensions are employed as the region features. This combination makes the man-made structure extraction in natural scenes more robust.

Keywords: Man-made structure extraction, Gaussian Mixture Model (GMM), Multi-Scale Random Field (MSRF), Tree-Structured Belief Network (TSBN), fractal dimension, multifractal estimation.

1 Introduction

The aim of this work is to extract man-made structure (specifically buildings) from 2D images. Our motivation is actually to assist in the segmentation of laser scan data (we have a system that collects laser scans and images of a scene with the latter to provide the colour information) so that, having separated the buildings from the other data, we can fit geometric models to the buildings. However, applications of building detection in 2D images are much wider - including image understanding.

For our particular setting, one can, of course, consider classification based on the 3D data itself, alternatively on the 2D data alone; or indeed a combination - that is, classification based simultaneously on the 2D image data and the 3D data. Angelov [1] is a recent example of the 3D data classification (classifying laser scan data into ground, building, tree or shrub). However, the results do not look particularly impressive. We know of no examples of the hybrid 2D-3D approach.

A considerable body of work in man-made structure extraction from 2D (image data) exists. In [2] a technique was proposed to learn the parameters of a large perceptual organization using graph spectral partitioning. However, these techniques require the low-level image primitives to be computed explicitly, and to be relatively noise-free. Oliva and Torralba [3] obtained a low-dimensional holistic representation of the scene using principal components of the power spectra. But the power spectra related assumption is not suitable for our images which contain a mixture of both the landscape and man-made regions within the same image. Closer to our approach, Hebert and Kumar [4] (following Bouman and Shapiro [7])

proposed a hybrid method which uses the bottom-up approach of extracting generic features from the image blocks, followed by the top-down approach of classifying image blocks based on the statistical distribution of the features learned from the training data. This Multi-Scale Random Field (MSRF) method yields better results compared with other approaches.

The problem we observed with the current supervised learning based segmentation, is that it cannot build a model accurate enough. There always exists an overlap between man-made structures and other structures/scene classes, where the model cannot distinguish one class absolutely from another class. This leads to false detection. Motivated by the observation that the world that surrounds us, except for man-made environments, is typically formed of complex and rough surfaces for which fractal provide a good model, our solution employs a fractal feature based rough segmentation (prior to further extraction using a simplified version of Kumar [4]).

The principal advantage of describing natural textures in terms of fractal surfaces is that it captures a simple physical relationship that underlies the texture structure and provides an accurate image segmentation procedure that is stable over a wide range of scales [6].

By segmenting the natural scene regions using a fractal model prior to supervised segmentation, the misclassification rate caused by the overlap between classes of the trained model is lowered.

Our approach is summarized in Figure 1, the input image data is processed in two ways. First, fractal features are calculated followed by a Fuzzy C-Means (FCM) clustering which roughly divides the image

Table 1. Comparison of previous approaches.

Authors	Data	Features	Classifier
Anguelov et al. 2005 [1]	3D	Local plane, points in a local cylinder, height	Associative Markov Network (AMN)
Kumar 2005 [5]	2D	3-scale analysis of edge orientation (interscale and intrascale)	Conditional Random Field (CRF)/ Discriminative Random Field (DRF)
Kumar and Hebert 2003 [4]	2D	3-scale analysis of edge orientation (interscale and intrascale)	Multi-Scale Random Field (MSRF)
Our work	2D	Intrascale features of Kumar 2005 + fractal measure	MSRF and Gaussian Mixture Model (GMM)

into regions with homogeneous fractal features. Second, features that reflect the gradient pattern of man-made structures are extracted. Then, by combining the FCM result with the MSRF model built from prior training data, a final segmentation extracts the building regions from the image. Table 1 contrasts our approach with previous ones.

The remainder of the paper has the following structure. In Section 2, an overview is given on the supervised segmentation as well as a description of the MSRF model and the Tree-Structured Belief Network (TSBN). Section 3 provides details on multifractal dimension and related feature estimation. In Section 4, results are presented concerning MSRF segmentation and of combining this with fractal clustering. Section 5 summarizes the main conclusions of the work and discusses possible future extensions.

2 Supervised Learning Segmentation

2.1 Overview

The input image is divided into non-overlapping 16x16 pixels blocks, and segmentation results in

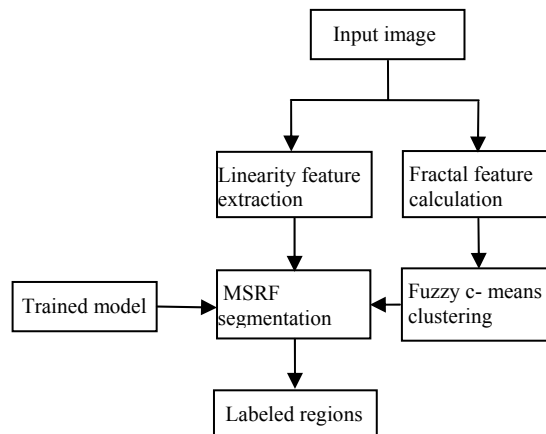


Figure 1. Overall approach to man-made structure extraction from natural scene.

labelling each block either as structure or non-structure class.

Kumar’s MSRF approach [4] is used for our supervised learning based segmentation.

This is formulated in the Bayesian framework, using a prior model (being a MSRF model in our application) which represents our knowledge about the label patterns; and a likelihood function to relate the image data to the class labels. A Gaussian Mixture Model (GMM) [8] is used for this likelihood function.

The classification problem is then interpreted as finding the optimal class labels which are obtained by maximizing the posterior probability over all image blocks.

The operation involves training and inference. Model parameters are learned through training and used for inference to yield the final segmentation.

2.2 Multi-Scale Random Field Model

The MSRF model is a better approach to Bayesian image segmentation compared with the fixed scale Markov Random Fields (MRF) model. It uses a pyramid structure to capture the characteristics of image behaviour at various scales. This is of critical importance since scale variation occurs naturally in images, and is important in quantifying image behaviour.

The fundamental assumption of a MSRF model is that the sequence of random fields from coarse to fine scale form a Markov chain. The random field at each scale is causally dependant on the coarser scale field above it. The Markov chain structure facilitates straight forward methods for efficient parameter estimation.

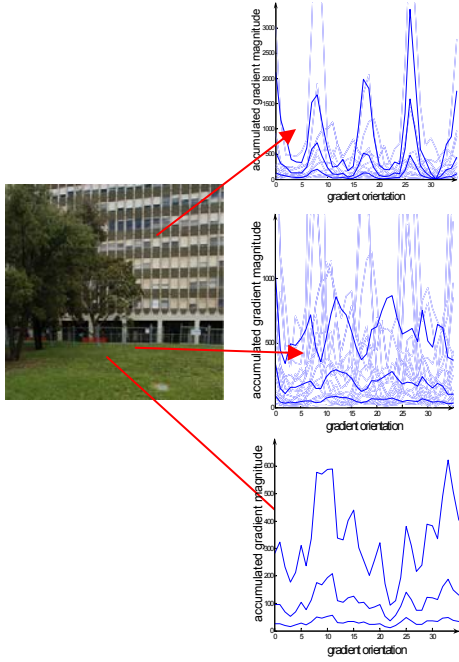


Figure 2. Orientograms computed at three sample points. For each point, three orientograms are derived at three different scales.

It is further assumed that points in each field are conditionally independent given their coarser scale neighbours. This leads to a rich model with computationally tractable properties.

Also, since explicit parameters (transition probabilities) are available to control both coarse and fine scale behaviour, the MSRF model can more accurately describe image behaviour.

An effective graphical representation of the dependencies embedded in our MSRF probabilistic models is a hierarchical structure known as Tree-Structured Belief Network (TSBN) [9]. In our work, training and segmentation inference are all done via Pearl’s message passing schemes on the TSBN tree [9].

2.3 Features

Features are computed at each block instead of at each pixel because integration over image blocks is necessary to compute the more complex features required for region classification. A feature vector is generated for each 16×16 pixel block independently.

These features attempt to capture the gradient pattern of the lines and edges in man-made structures, as opposed to the less structured characteristics in natural objects. This is accomplished using a set of features derived from histograms of gradient

orientations in a region weighted by gradient magnitudes [10] - termed “orientograms” - generated at three different scales; 1×1 , 2×2 , and 4×4 blocks. The gradient magnitudes are calculated using a derivative of Gaussian filter (scale around 5 pixels is typical in our implementation). As can be seen in Figure 2, there are clear extrema of orientation for buildings, while for tree and ground, the peaks are more randomly distributed.

In order to alleviate the hard binning of the data, the histogram is smoothed using kernel smoothing as follows,

$$E'_\delta = \frac{\sum_{i=1}^{\Delta} K((\delta - i)/h) E_i}{\sum_{i=1}^{\Delta} K((\delta - i)/h)} \quad (1)$$

where E_δ be the magnitude of the orientation histogram at the δ^{th} bin, Δ be the total number of bins in the histogram and K is the Gaussian smoothing kernel function with bandwidth h . The bandwidth of the kernel is chosen to be 0.7 to restrict the smoothing to two neighbouring bins on each side.

We use the mean magnitude of the orientogram at three different scales as features in our MSRF model.

3 Unsupervised Segmentation with Multifractal Dimension

Fractal dimension is a mathematical idealisation, while real world textures are only “semi-fractals” that have anisotropic and inhomogeneous scaling properties which can not be well characterized by the fractal dimension. In order to obtain a satisfactory texture analysis, a set of measures, instead of one single measure, should be used - i.e., multifractal analysis [11].

There are several ways available to estimate the multifractal dimensions of the image. Instead of the most common box counting method, we use the morphology based estimation proposed by Xia, Feng and Zhao [11] which is more straightforward and accurate.

An $M \times N$ image is considered to be a 3-D surface X , which can be defined as a set of triplets $\{(i, j, f(i, j)); i = 1, 2, \dots, M; j = 1, 2, \dots, N\}$

A series of cubic structure element (SE) of different scales is used to measure the image surface. For every scale ϵ , the SE Y_ϵ is also given as a set of triplets $\{(i_{\epsilon k}, j_{\epsilon k}, \beta_\epsilon); k = 1, 2, \dots, P_\epsilon\}$, where P_ϵ is the number of elements in Y_ϵ and β is the SE shape factor. The dilation of X with Y_ϵ at pixel (i, j) is calculated as

$$f_\varepsilon(i, j) = \max_{k=1,2,\dots,P_\varepsilon} \{f(i + i_{\varepsilon k}, j + j_{\varepsilon k}) + \beta\varepsilon\} \quad (2)$$

A local natural measure $\mu_\varepsilon(i, j)$ is defined in size $W \times W$ window as

$$\mu_\varepsilon(i, j) = \frac{|f_\varepsilon(i, j) - f(i, j)|}{\sum_{i,j}^W |f_\varepsilon(i, j) - f(i, j)|} \quad (3)$$

The measure of order q at scale ε can be calculated as

$$I(q, \varepsilon) \equiv \alpha \sum_{i,j}^W \mu_\varepsilon(i, j)^q \quad (4)$$

where

$$\alpha = \frac{\sum_{i,j}^W |f_\varepsilon(i, j) - f(i, j)|}{\varepsilon} \quad (5)$$

Being a multifractal measure, $I(q, \varepsilon)$ must satisfy the following power law:

$$I(q, \varepsilon) \sim \varepsilon^{\tau(q)}, \quad -\infty < q < \infty \quad (6)$$

Where $\tau(q)$ is the multifractal dimension spectrum. Then, the local morphological multifractal exponents (LMME), are defined as follow:

$$L_q = \frac{1}{|q|} \lim_{\varepsilon \rightarrow 0} \frac{\ln(I(q, \varepsilon))}{\ln(\frac{1}{\varepsilon})}, \quad q \neq 0. \quad (7)$$

RANSAC is used to fit a line to the group of data $(\ln(I(q, \varepsilon)), \ln(1/\varepsilon))$, calculated at a given set of scales, from which the limit in (6) can be estimated.

On each pixel, several values of the LMME spectrum are calculated, forming a LMME vector on which fuzzy C-means clustering is applied to roughly divide the image into several regions with homogeneous fractal character. Among these regions, the region(s) with small LMME value, which corresponds to low roughness of man-made structure surfaces, are picked out for further supervised segmentation.

4 Experiments and Results

4.1 MSRF Segmentation

The proposed algorithm was trained and tested using images that Kumar used, which are available from <http://www.cs.cmu.edu/~skumar/manMadeData.tar>. We used 93 images and the corresponding labels, with the size of all images being cut to 256x256.

The training images are divided into non-overlapping 16x16 pixels blocks which are labeled as one of the two classes, i.e. building or non-building blocks.

A GMM model is trained for each of the two classes. Bouman's cluster program, which can be downloaded from <http://cobweb.ecn.purdue.edu/~bouman/>, is used for the GMM parameter estimation. The program applies the expectation maximization algorithm together with an agglomerative clustering strategy, using minimum description length to estimate the number of clusters which best fits the data.

In order to learn the parameters of the MSRF, the image size has been cut to 256x256 and a 5 level quad-tree is built considering each 16x16 pixels non-overlapping block in the image to be the leaf node (level 5). The MSRF model here is a two class model: building (B) and non-building (N). The initial parameter values are obtained by building the empirical trees over the image labels in the training images using max-voting.

Figure 4 shows two test images together with its GMM and MSRF classification result. It can be seen that by adding MSRF, the false detection rate is obviously reduced. MSRF model tends to smooth the labels in the image and removes the isolated false detections.

4.2 Adding Multifractal Clustering

The MSRF model does not work well if there exist objects and scenes with complex and rough surfaces such as vegetation and hills. It can be seen from Figure 5 that the MSRF model cannot discriminate the rough parts of a scene from buildings. By adding the fractal based clustering, the accuracy of the extraction is greatly enhanced.

The multifractal estimation is performed at each pixel (i, j) using a moving window size of 11 x 11 centered on (i, j) . The SE shape factor β is 3, and the scales ε of SE are 2, 3, 4, 5, and 6, respectively. Three components of the LMME spectrum, i.e. L_{-2}, L_{-1} and L_1 are used as features for fuzzy C-means clustering with the results shown in (e) and (f) of Figure 3.

As shown in Figure 3 by sequentially increasing the number of clusters, we can detect a "knee" in the objective function (essentially the approximation error) where the change of the object function value starts to slow down. This is used to choose the number of clusters for the segmentation.

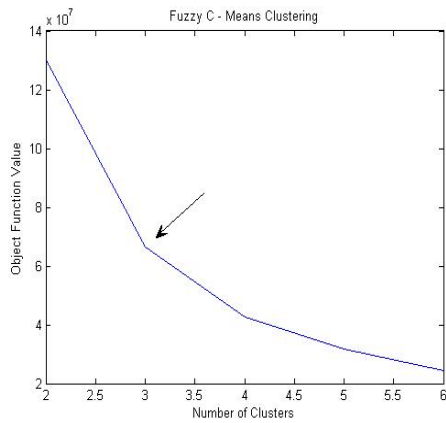


Figure 3. Selection of cluster numbers.

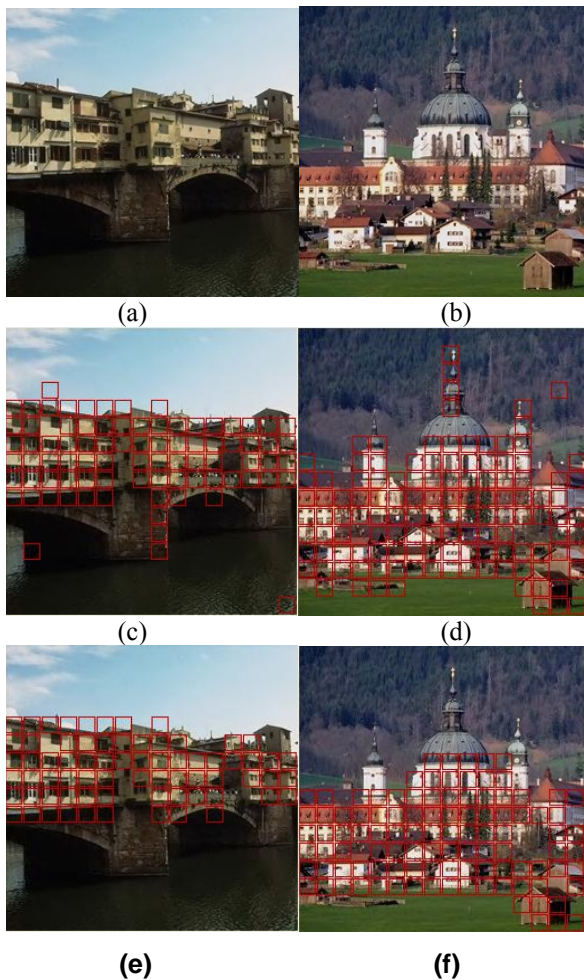


Figure 4. MSRF segmentation results. (a) (b) original images, (c)(d) GMM results, (e)(f) GMM+MSRF

For the segmented regions, the one that covers the building needs to be identified and used for further MSRF classification, e.g., (e) and (f) in Figure 5. Since the LMME value of each pixel is proportional to the roughness of the image, a threshold (set to be 0.1) is used to filter out pixels whose values are below this threshold (relatively "flat" image sites - typical of buildings). Then within each region, we calculate the non - fractal proportion of such pixels in those regions, and the region(s) with non - fractal proportion greater than a threshold (being 0.3 here) are selected for MSRF analysis (final detection for buildings).

Table 2 show example percentages for the data in Figure 5 (f). It can be seen that Block '4' which covers the building obviously has a higher non - fractal proportion value.

Table 2. Locating the right block

Block index	Non - fractal proportion
1	19%
2	7%
3	9%
4	34%
5	21%
6	17%

5 Conclusions and Future Work

In this paper, we have proposed a new method for segmentation of man-made structure from natural scene. The key novelty of our algorithm is combining the Bayesian network (MSRF) based supervised segmentation with unsupervised fractal segmentation. The latter helps to remove false positives generated by vegetation, and thus effectively lower the misclassification rate.

Compared with the conventional supervised segmentation, less features and training images are required to achieve the same results.

However, several issues need to be addressed. The current method can only detect buildings that fill a significant (and connected) portion of the whole image. A more sophisticated solution needs to be developed to extract relatively small and scattered buildings in the image. In the meantime, the features of the MSRF model can be further optimized. Furthermore, a hybrid of 2D-3D approach can be investigated for our laser scan data segmentation by taking advantage of the accurate geometric features of the 3D data.

6 References

- [1] D. Anguelov, B. Tasker, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Ng, "Discriminative Learning of Markov Random Fields for Segmentation of 3D Scan Data," in *CVPR2005*, San Diego, 2005, pp. 169-176.
- [2] S. Sarkar and P. Soundararajan, "Supervised learning of large perceptual organization: Graph spectral partitioning and learning automata," *IEEE Trans. on Pat. Anal. Mach. Intell.*, vol. 22, pp. 504-525, 2000.
- [3] A. Oliva and A. Torralba, "The Shape of the Scene: a Holistic Representation of the Spatial Envelope," *Intl. Journal of Computer Vision*, vol. 42, pp. 145-175, 2001.
- [4] S. Kumar and M. Hebert, "Man-made structure detection in natural images using a causal multiscale random field," in *CVPR2003*, 2003, p. 119.
- [5] S. Kumar, "Models for Learning Spatial Interactions in Natural Images for Context-Based Classification," The Robotics Institute, School of Computer Science, Carnegie Mellon, Pittsburgh, PhD thesis 2005.
- [6] A. P. Pentland, "Fractal based description of natural scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 661--674, 1984.
- [7] C. A. Bouman and M. Shapiro, "A Multiscale Random Field Model for Bayesian Image Segmentation," *IEEE Transactions on Image Processing*, vol. 3, pp. 162-177, 1994.
- [8] G. McLachlan, *Finite Mixture Models*: Wiley, 2000.
- [9] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*: Morgan Kaufman, 1988.
- [10] W. A. Barrett and K. D. Petersen, "Houghing the hough: Peak collection for detection of corners, junctions and line intersections," in *CVPR2001*, 2001, pp. 302-309.
- [11] Y. Xia, D. Feng, and R. C. Zhao, "Morphology-based multifractal estimation for texture segmentation," *IEEE Transactions on Image Processing*, vol. 15, pp. 614-623, 2006.

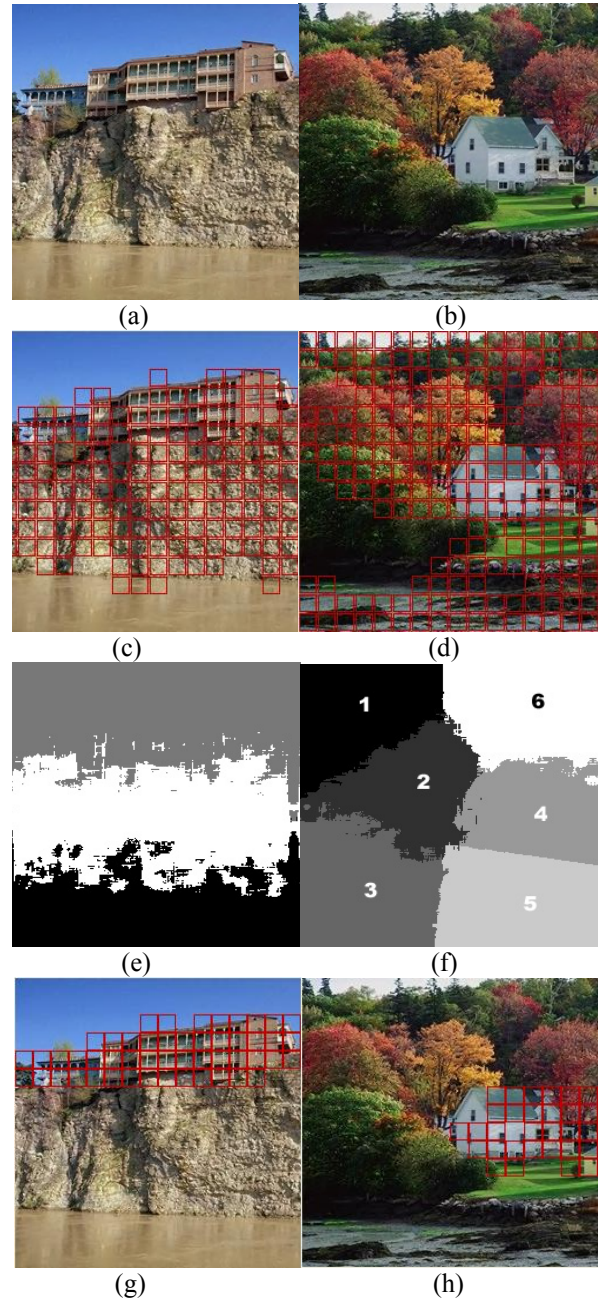


Figure 5. Fractal+MSRF segmentation results. (a)(b) original images, (c)(d) GMM+MSRF results, (e)(f) fractal based clustering (g)(h) Fractal+MSRF segmentation.

Accelerating calibrated stereo correspondence through concurrent processing

Nathan Adams and Richard Green

CSSE, University of Canterbury

Email: nad37@student.canterbury.ac.nz, richard.green@canterbury.ac.nz

Abstract

We examine the availability of processing power for stereo disparity matching in an increasingly parallel computing environment. We examine the application of this trend with regard to the stereo correspondence problem, and particularly the set of conditions imposed by the use of USB webcams. We discuss an example correspondence algorithm and how to parallelize it, and experimentally demonstrate a framerate increase on a dual-core CPU with an existing implementation that reaches what can be considered real-time performance in the webcam scenario, achieving almost complete utilization of available processing cores.

Keywords: stereo vision, stereo correspondence, webcam

1 Introduction

The problem of determining disparity correspondence between two (or more) cameras has been a subject of research for a long time. The stereo correspondence problem is one that is hard to produce results of reasonable quality in sufficient time, an ideal goal being a system that is capable of processing stereo image pairs in realtime as they are captured by cameras.

Helping to combat this has been the continual improvement of processor performance, with hardware now at the point where realtime performance is possible for many algorithms. Particularly promising has been the use of programmable GPUs for this and other computer vision tasks.

In this paper we examine the growing utility of parallelizing stereo-matching algorithms, and the performance of a simple and almost algorithm-independent approach to parallelization of the correspondence problem, to obtain real-time performance in an environment powered by normal desktop or laptop hardware.

2 Background

Recent work such as that by Woetzel and Koch [1], Gong and Yang [2], Yang et al. [3], and Fung and Mann[4] has shown the utility of modern graphics cards for running computer vision applications such as stereo correspondence.

Fung and Mann presented the OpenVIDIA library[5], implementing several computer vision

in Cg and OpenGL to run on NVIDIA graphics cards. OpenVIDIA is currently available with a demonstration of an implemented stereo correspondence algorithm (based on a planar sweep), which, with an NVIDIA GeForce 6600GT can process the ‘map’ Middlebury dataset[6] in 5.2 ms. That this performance can be achieved independently of the CPU speaks of the merits of off-loading such tasks to the GPU.

The task of programming for GPUs is also being made easier with such projects as Stanford’s BrookGPU[7], which uses C++ templating to allow programmatic access to the GPU in a high level fashion. Sh[8] is a similar effort, aimed at providing a streams-based metaprogramming language that can compile to both CPU and GPU backends. This adds to the attractiveness of using the GPU in stereo correspondence and other computer vision tasks.

GPUs are not the only alternate processing power becoming more prevalent in the mainstream market. Increasingly the retail CPU space is being occupied by processors with two or more cores, presently by chips such as Intel’s Core 2 Duo, with four core processors on the upcoming product schedules of both AMD and Intel. As the industry is expected to follow a trend of adding more cores to processors, rather than increase clock-cycles alone, increasingly there will be a proportionately wasted amount of available processing power if algorithms that block further computation are not making full use of all available cores. In the realm of computer vision, stereo correspondence is often such a blocking activity, as there can be many

operations in a stereo vision system’s pipeline that depend upon a depth map of the scene presented to the cameras.

From these trends we can extrapolate an importance requirement of future stereo correspondence algorithms. Increasingly, the hardware power will be available to achieve real-time or better performance, but it may well be spread across multiple processing cores, whether they be on the same processor, or in varying configurations of differing numbers of GPUs and CPUs. The challenge will be to make full use of this power in a scalable manner.

3 Threading an existing algorithm

3.1 Single-thread performance check

We wrote a test harness using standard functions from the Intel OpenCV library[9]. Using two Logitech QuickCam Pro 5000 USB webcams in a rough stereo configuration, the harness performs intrinsic and extrinsic camera calibration with video of the user moving a standard chessboard calibration pattern[10]. An example of this process is shown in Figure 1. It then establishes from these the cameras’ epipolar geometry in the form of the fundamental matrix.

Pre-calibrating the cameras like this holds important advantages. Extraction of the fundamental matrix allows rectification of subsequent image pairs (that is, adjusting the images so that the epipoles of the image pair align with their scanlines). As we see later, this allows for greater ease of parallelization, if the algorithm is of a class that requires rectified images. Even if the chosen correspondence algorithm does not have any such requirement, the calibration is required anyway. Webcams provide poor quality images even in the best of conditions. Often this means they suffer from radial lens distortion, and so whether or not the stereo correspondence algorithm requires it, calibration is a necessary step. Additionally, due to internal exposure mechanisms that are usually uncontrollable in software, even pre-calibrated webcams can require recalibration if the lighting conditions change too much.

The harness uses the established epipolar geometry and the (as of version 0.97) experimental OpenCV function `cvFindStereoCorrespondence`[11] to produce a depth image from the stereo images. The algorithm used is based upon Stanfield and Tomasi’s dynamic programming algorithm [12], with a modified cost function.

For the purposes of performance timing, the harness was given pre-recorded stereo video as cali-

bration material, and then played the same videos again in the stereo matching phases to ensure calibration was correct across trials.

The test machine was configured with an Intel Core 2 Duo E6400 (clocked at 2.13GHz) and 1GB of RAM, running Windows XP Professional. The test videos were recorded at 320 by 240 pixels in size. In the initial test, this yielded an average performance of 117 ms per frame.

3.2 Threading

Ideally on a computer with a dual-core CPU this processing time could be halved by utilizing the other core, and more generally with scalability on the order of $O(np^{-1})$, where p is the number of processing cores available. The previously noted suitability of the GPU for the task of stereo matching suggests the task is well-adapted to parallel processing. This leads to the question of implementation. In the case of a dense algorithm (ie, every pixel in the depth image is independently calculated, rather than interpolated from significant points in the image pair as it may be in a feature-based algorithm) such as the one used in `cvFindStereoCorrespondence`, there are only really two possible divisions of labour to thread the system with: time and space.

3.2.1 Dividing labour over time

In the case of this algorithm, one way to split workload across time would be to split the algorithm into two basic parts: finding the corresponding points in a scanline pair, and post-processing data column-wise between scanlines. The fundamental flaw with this approach is the asymmetry of the tasks, which is an algorithm-independent problem with this approach. Since the processor cores are of identical speed, the elementary answer is that they are best utilized by sharing identical tasks. This also aids scalability of the algorithm as more processing cores are added. An example of a naïve implementation would be dedicating sequential image pairs to the processing cores in a round-robin fashion, so that frame pair f is assigned to be completely processed by core c in a system with n cores, where

$$c = f \pmod{n} \quad (1)$$

This approach, while theoretically reducing the average time to process an image pair by a factor of n , has two pathological downsides. The first is that even if the cores can maintain an average framerate that is sufficient to call realtime (ie, in this case of stereo USB webcams, 15 frames per second),

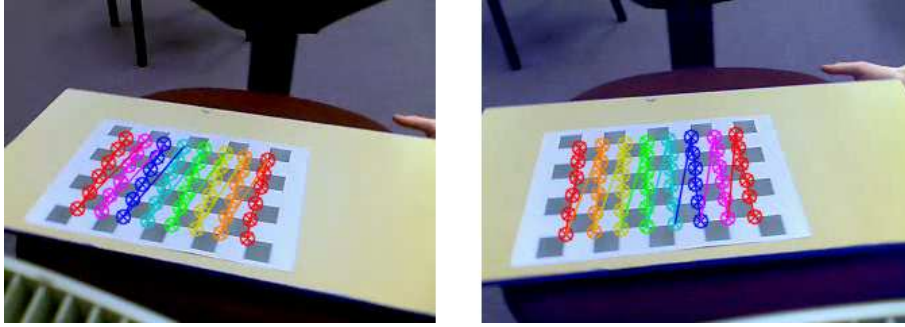


Figure 1: Views from the stereo webcams in an example calibration, with chessboard corner recognition markers overlaid.

there can still be a latency of up to $n - 1$ frames between image-pairs entering the stereo-matching algorithm, and their corresponding disparity map emerging. The second and more likely problem is that this leads to a stuttering framerate as the cores risk falling into a pattern of rapidly accepting n queued input image-pairs and processing them while more image-pairs queue, awaiting processing.

A third problem — albeit algorithm-dependent — arises from the situation where the algorithm depends on knowledge of the previous frame(s), a simple example being where the previous image pair is subtracted from the current one to cut down on the amount of pixels that have to have their disparities recalculated. In this case, the system collapses back to the performance of a single core as each image pair waits upon the completion of the previous.

3.2.2 Dividing labour over space

The other approach to multithreading stereo-matching, division of labour over image-space, turns out to be much more practical for most algorithms. Particularly well suited to this are the dense grey-matching algorithms. As mentioned earlier, the input images to the `cvFindStereoCorrespondence` function are rectified, thus the need for calibration at the start of the session. The advantage of this in this algorithm's case is that the most time-consuming phase, matching points between scanline pairs and assigning costs to each, is highly parallelizable. The point matching calculations to be performed on each scanline pair can be done so completely independently of each other. The only algorithmic limit on concurrency here is the number of scanlines.

The situation changes after that step. The function `cvFindStereoCorrespondence` then enters a column-wise post-process, propagating information between scanlines. Theoretically

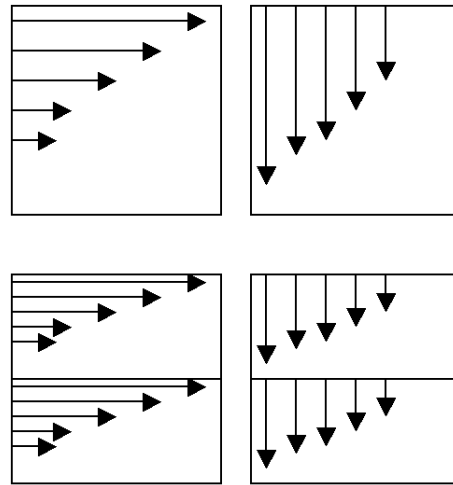


Figure 2: Top: Single-thread processing an image (one of a pair). Bottom: An image being process by two threads. Arrows indicate direction of data reading and writing. Note the severed column-wise data flow in the two-thread example.

this process is just as parallelizable as that of the scanline pair feature matching. However, splitting column-wise across the threads at this point requires all the threads wait after the first part of the algorithm. Depending on the platform (specifically in a scalable system, the latency between computing nodes, whether they be individual CPU cores, individual CPUs, or even separate computers linked over a network), it may be desirable to avoid any such time overhead. With an algorithmic-dependent compromise in result quality (since various algorithms carry out different numbers of horizontal and vertical passes), we can continue the algorithm still maintaining the same allocation of image space to each thread. That is, in the case of two threads, the first thread is allocated the top half of each image in the pair, and the second thread the bottom (see Figure 2). This part unfortunately is not scaleable, since as n increases, the size

of the vertical passes decreases, leading to the deterioration of results.

Another advantage of only separating the workload across threads once is that it allows the programmer to leave the original algorithm more intact if it is being parallelized post-creation, as we did to `cvFindStereoCorrespondence`.

4 Results

Taking the approach described only required approximately 25 lines of C++ code to parallelize `cvFindStereoCorrespondence`. When benchmarked, it gave the results seen in Table 1.

Table 1: Results (Core 2 Duo E6400 2.13GHz)

Threads	Average framerate
One	8.544
Two	16.204

Given the physical limits of the USB webcams often allow no more than 15 frames per second, it is arguable that this parallelization has brought the system into the realm of real-time. The system frame-rate has also nearly doubled, indicating the second core is now being utilized almost completely efficiently.

5 Conclusion

In this paper we examined the availability of processing power for stereo disparity matching in an increasingly parallel computing environment. We examined a pre-existing algorithm performing at sub-real-time speed on a modern system, and examined it for parallelization, rejecting implementations as inappropriate on the basis of scalability, overly algorithm-dependent attributes, and other grounds. Upon implementation, a simple image-space splitting approach was able to garner close to complete utilization of the second core. As multiple core processors become more prevalent in the mainstream market, this technique could easily be applied to other stereo-matching algorithms with suitable success.

5.1 Limitations and future work

As mentioned earlier in the paper, GPUs have been proven to perform well in this area. This paper does not cover the implementation and synchronisation complexities of integrating of GPUs into such a system. Future research is possible in the area of developing frameworks that will scale stereo matching algorithms across CPU cores and GPUs alike.

References

- [1] J. Woetzel and R. Koch, "Real-time multi-stereo depth estimation on GPU with approximate discontinuity handling," *Visual Media Production, 2004.(CVMP). 1st European Conference on*, pp. 245–254, 2004.
- [2] M. Gong and Y. Yang, "Near real-time reliable stereo matching using programmable graphics hardware," *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, 2005.
- [3] R. Yang, M. Pollefeys, and S. Li, "Improved Real-Time Stereo on Commodity Graphics Hardware," *Computer Vision and Pattern Recognition Workshop, 2004 Conference on*, pp. 36–36, 2004.
- [4] J. Fung and S. Mann, "OpenVIDIA: parallel GPU computer vision," *Proceedings of the 13th annual ACM international conference on Multimedia*, pp. 849–852, 2005.
- [5] Eyetap Personal Imaging Lab, "Openvidia : Parallel gpu computer vision." <http://openvidia.sourceforge.net/>, visited on 11/9/2006.
- [6] Scharstein, D. and Szeliski, R., "Middlebury college stereo vision research page." <http://cat.middlebury.edu/stereo/data.html>, visited on 10/9/2006.
- [7] Stanford Graphics Lab, "Brookgpu." <http://graphics.stanford.edu/projects/brookgpu/>, visited on 7/9/2006.
- [8] M. McCool and S. Du Toit, *Metaprogramming GPUs with Sh*. AK Peters, 2004.
- [9] Intel Corporation, "Open source computer vision library." <http://opencvlibrary.sourceforge.net/>, last visited on 11/9/2006.
- [10] Z. Zhang, "A flexible new technique for camera calibration," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 11, pp. 1330–1334, 2000.
- [11] "OpenCV documentation." <http://opencvlibrary.sourceforge.net/CvAux>, last visited 11/9/2006.
- [12] S. Birchfield and C. Tomasi, "Depth Discontinuities by Pixel-to-Pixel Stereo," *International Journal of Computer Vision*, vol. 35, no. 3, pp. 269–293, 1999.

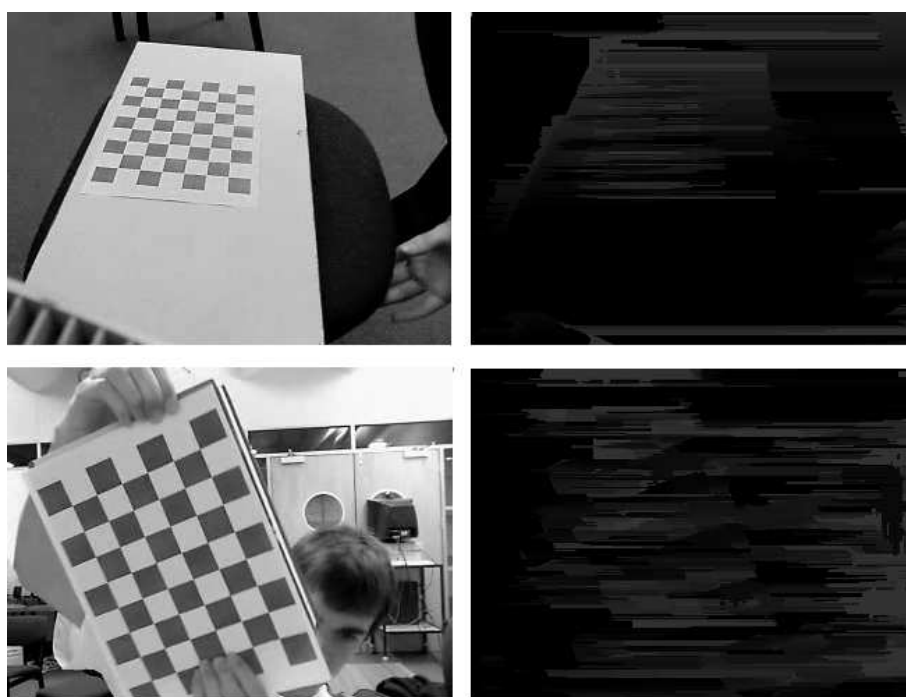


Figure 3: Left images: View from the left camera. Right: The corresponding depth image calculated for that frame.

Local Texture Patches for Active Appearance Models

N. Faggian¹, A.P. Paplinski¹ and J. Sherrah²

¹Monash Univeristy, Clayton School of Information Technology, Melbourne, Australia.

²Clarity Visual Intelligence Pty Ltd, Melbourne, Australia.

Email: nathan.faggian@infotech.monash.edu.au

Abstract

This paper presents a new fitting strategy for Active Appearance Models (AAMs.) The warping function is implemented using orientated patches and is based on the theory of Active Shape Models (ASMs). We present a modified inverse compositional fitting strategy to fit our patch-based AAM (PAAM). The modification allows the fitting of the PAAMs with similar accuracy to AAMs but using far less texture information. We compare our implementations of the AAM against the PAAM using three different data sets. Through our experiments we show that there is almost no additional error introduced by our reduced local texture models.

Keywords: Active Appearance Models, Active Shape Models

1 Background

Active Appearance Models (AAMs) are a popular technique to model objects in images. It is a two stage modeling-by-synthesis approach that has been broadly used in the field of computer vision and was first introduced by Cootes et al [1]. An AAM encodes both the shape and texture information of an object in an image and is typically built using hand-labelled data. The first stage of an AAM is training, where corresponding features (which form shapes) are labelled across an image set and then aligned using Procrustes [2] alignment. Once the shapes are aligned, the texture and shape vectors are stacked into matrices and Principal Component Analysis [3] (PCA) is performed. Here the goal is to estimate any valid instance of the object using PCA as the parametric models for shape and texture variation:

$$\hat{s} = \bar{s} + S \cdot \text{diag}(\sigma) \cdot \gamma \quad (1)$$

$$\hat{t} = \bar{t} + T \cdot \text{diag}(\sigma) \cdot \Omega \quad (2)$$

The shape and texture models encode the modes of variation that the labeled image set provide. A new shape \hat{s} or texture \hat{t} can then be constructed as a linear combination (γ or Ω) of the principle components (column space) of the measurement matrices for shape and texture. These modes of variation are shown in figure 1, where the first principle component of both the shape and texture models are varied from -3.5 to 3.5 square roots of the first eigenvalue.

Active Shape Models (ASMs) are similar two stage models and were also introduced by Cootes et al [4]. The key difference between the ASM and the AAM is simply the size of the much smaller texture model in the ASM. This is due to the ASM only modelling pixels along the normal to shape vertices. ASMs have been

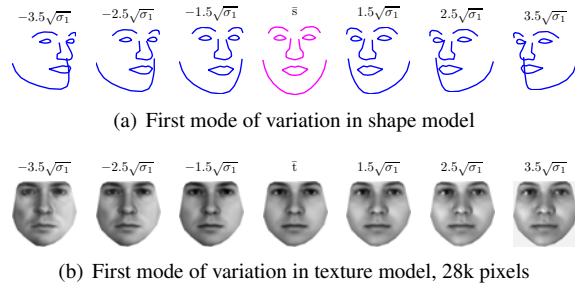


Figure 1: AAM: Shape and texture models

applied extensively in computer vision also, however the AAM demonstrated that their performance could be improved by modelling the complete texture of the object. The final stage of the AAM and ASM is fitting. In this paper we have taken the concept of the ASM and the AAM and combined them to reduce texture model sizes and increase the efficiency of a popular AAM fitting method (ICIA) [5].

2 AAM Fitting

Fitting the texture and shape models for an AAM is a complex nonlinear problem. This is because pixels and their locations in images are generally not related [6]. Fitting an AAM is much like the nonlinear optimization applied to image alignment. In fact image alignment algorithms can be directly applied to the fitting of an AAM. In this scenario the AAM (shape component) is represented as a special transform and is called a piecewise affine transform. This paper demonstrates that the common use of the piecewise transform is not required for fitting AAMs.

Image alignment is generally a nonlinear optimization task that is commonly solved using Gauss-Newton op-

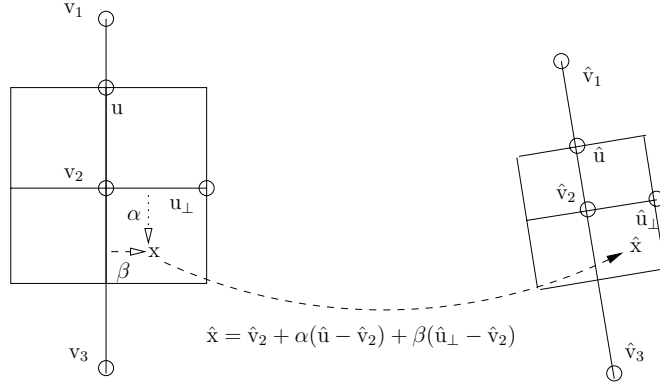


Figure 2: Transforming a pixel, x , in patch defined by v_1, v_2, v_3 to its corresponding position, \hat{x} , in a different patch defined by $\hat{v}_1, \hat{v}_2, \hat{v}_3$.

timization [3]. This was first proposed by Lucas and Kanade [7] to solve for the affine relationship between an image and a transformed template. The approach was later extended to AAMs by Baker et al [5, 8, 9], in a method termed Inverse Compositional Image Alignment (ICIA). In this paper we extended ICIA to use a modified sampling transform that reduces the size of the texture model required for AAM construction and fitting.

3 Modifying ICIA AAM Fitting

ICIA fitting is an analytical solution to fitting AAMs which is presented in an optimization framework. ICIA minimizes the cost function:

$$\epsilon = \sum_x [A_0 - I(W(x; \gamma))]^2 \quad (3)$$

where A_0 is the mean shape and mean texture AAM render and $I(W(x; \gamma))$ is the image sampled to the mean shape using the transforming parameters γ . Here γ is simply the shape parameters for the AAM shape model. The key to ICIA fitting is to swap the template and the image during optimization, which makes the transform Jacobian $(\frac{\partial W}{\partial \gamma})$ constant.

There has already been a substantial amount of work that extends ICIA to the task of AAM fitting [5, 8, 9]. This paper demonstrates how to remove the standard piecewise transform and replace it with a patch-based transform in ICIA. To do so we need to modify the ICIA algorithm presented in [6]. In this section we define the key differences presented by our method. Specifically we will now define: 1) A modified patch-transform function and 2) A different Jacobian structure.

3.1 The Patch-Transform, $W(x; \gamma)$

A transform changes the pixels x from one shape into another using the parameters γ ; $W(x; \gamma)$. In order to define the transforming function $W(x; \gamma)$ we have to define the connectivity of the shape model. For our approach we define an oriented patch centered on each model vertex.

The linear algebra for our patch representation is based on that used in the ASM where the sampling at each vertex is done along the normal to that vertex. To do so, three connected shape model vertices are required. From these vertices's we can determine two vectors:

$$v_d = v_3 - v_1 \quad (4)$$

$$v_n = \begin{bmatrix} -v_d(2) \\ v_d(1) \end{bmatrix} \quad (5)$$

where v_1, v_3 are the adjacent vertices to the current model vertex, v_2 . With this information it is possible to define the vector from v_3 to v_1 , which is v_d and consequently the normal of v_2 which is v_n . This leads to the normal equation used in the ASM. Here the size of the sample (in pixels) is defined by a length k along the normal direction:

$$x = v_2 + k \cdot v_n \quad (6)$$

where v_2 is the model vertex of interest, and x is the sampled pixel at distance k . In our method we use a similar approach to the ASM for local texture modelling. Instead of a single line we have introduced the idea of a patch with a width and height defined by a patch size constant, K pixels. We define the patch as the combination of orthogonal vectors and call them the principle directions. To define the first principle direction we must define a single point that is K pixels along the normal to the current model vertex v_2 :

$$u = v_2 + K \cdot v_n \quad (7)$$

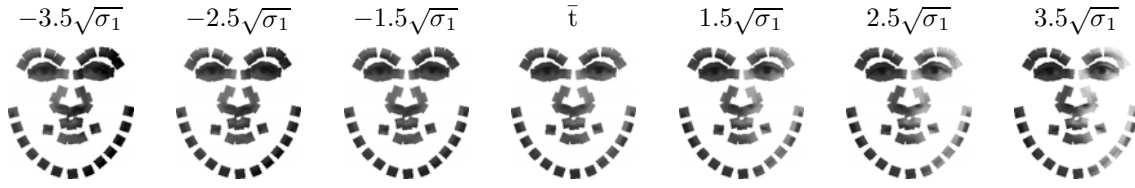
We must also defined another point along the orthogonal direction to u :

$$u_{\perp} = v_2 + K \cdot v_d \quad (8)$$

Using these two points it is possible to define all pixels in the patch using barycentric coordinates. Where a pixel, x , in the patch is defined as a magnitude along the direction from v_2 to u_{\perp} and from v_2 to u . An example is shown in figure 2. This leads to the transforming function that we use for patch fitting:

$$w(x; \gamma) = v_2 + \alpha(u - v_2) + \beta(u_{\perp} - v_2) \quad (9)$$

It is important to note the similarities to the equation of Baker and Matthews [5]. This is a mathematically



(a) First mode of variation in reduced texture model, 14k pixels and 50% smaller than the original AAM

Figure 3: AAM: Patch transformed texture model

useful form that allows the Jacobian of the transform, with respect to a model vertex ($\frac{\partial W}{\partial v}$), to be computed easily. The key difference in our approach is that we do not strictly adhere to the true barycentric form, allowing both α and β to be negative.

For ICIA AAM fitting the key task is to transform the image back into the AAM coordinate frame. This equates to transforming a shape, \hat{s} , built from equation (1) into the mean shape, \bar{s} . To do so with our modified patch-transform we require the computation of each principle direction per vertex and the corresponding barycentric coordinates. In practical terms the barycentric coordinates for the patches of the mean shape are cached. This means that to transform using our method only the principle directions for the new shape \hat{s} need to be computed. An example of this transform is shown in figure 3.

3.2 The Patch-Transform Jacobian

In the implementation of the AAM outlined in [5] model points are triangulated and the triangle Jacobians are summed together. We have removed the triangulation and in doing so have derived a simpler representation of the optimizations Jacobian for each patch, v .

$$\frac{\partial W}{\partial v} = 1 - \alpha - \beta \quad (10)$$

The Jacobian is derived from equation (9). Using our patch transform, for each vertex in the model there is an orientated patch, the angle of which is defined solely by the neighboring vertices. The orientation of the patch affects the Jacobian which is now a plane that has a saddle point on the vertex v . This Jacobian is multiplied with the Jacobian of the transform with respect to the shape model ($\frac{\partial W}{\partial \gamma}$). The new transform Jacobian is now a product of equation (1) and¹ the new definition for the patch-transform:

$$\frac{\partial W}{\partial \gamma} = \frac{\partial W}{\partial v} \cdot \frac{\partial W}{\partial \gamma} \quad (11)$$

3.3 Practical considerations

In this section we outline the required changes to ICIA AAM fitting for our patch-based transforming function.

¹ $\frac{\partial W}{\partial \gamma}$ is the Eigenvectors of the shape model in equation (1)

The standard AAM (with patch-transform) is presented in Algorithm 1. During implementation there are a number of practical as well as necessary additions that should be made to the standard algorithm outlined in [5]. We have implemented all of these additions in both our implementation of the AAM and the Patch-AAM (PAAM).

Algorithm 1 AAM Fitting

```

J = ∇A0  $\frac{\partial W}{\partial \gamma}$ 
{precompute the Jacobian}
γ = [0]
{set the initial parameters for shape}
while Δγ > 1e-4 do
  I(W(x; γ))
  {transform the patches to the AAM coordinate frame}
  ε = (A0 - I(W(x; γ)))
  {compute the error vector}
  H = (J) · (J)T
  {compute the Hessian}
  Δγ = H-1 · JT · ε
  {compute the parameter update}
  γ = γ ∘ Δγ-1
  {compose the update with the current parameters}
end while

```

The first addition is to include a global normalizing transform. This appends a similarity transform to the AAM fitting. In doing so we suggest that the local and global transforms are orthogonalized. This can be done using QR factorization and is outlined in [10]. We also suggest that the gradient function for computing image derivatives be a plane based method. In the AAM there are discontinuities at the model edges that should not be encoded.

The second addition would be to extend the method using the well known Levenberg-Marquadt optimization. This typically improves the convergence speed of the AAM fitting.

The third and final addition to improve generality could be to use the simultaneous ICA algorithm. It has been documented in [11] that person specific AAMs perform much better than AAMs trained using many different identities (generic AAMs). One solution is to update the computation of the transform Jacobians during fitting. This improves the generic models convergence

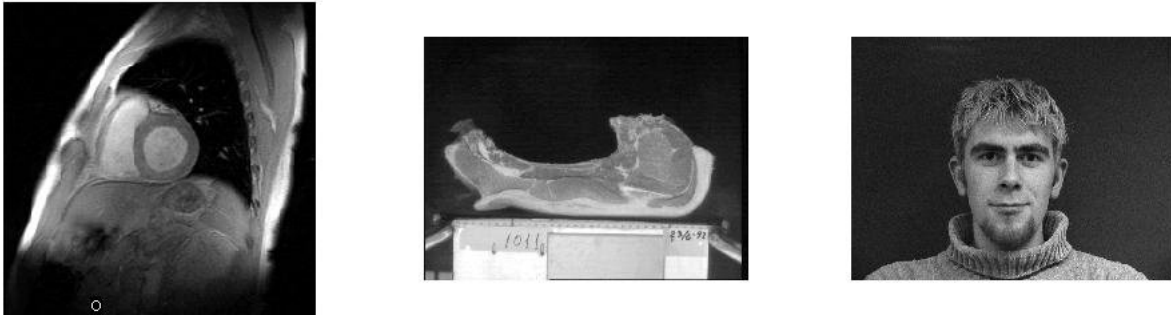


Figure 4: Ventricle, Meat and Face dataset examples.

properties and results in a slightly different (less efficient) algorithm called simultaneous ICA [11].

4 Experiments

For our modified AAM, experimental results are reported for three datasets: 1) 14 meat images with 83 landmarks [12], 2) 14 Ventricle images with 66 landmarks [13] and 3) 36 face images with 58 landmarks [14]. Each of the datasets provided a linkage matrix, describing landmark connectivity. Examples of the data are shown in figure 4.

The first experiment examined the generalization properties of the PAAM and AAM. To this end we performed a 4-fold cross-validation for all sets. Measuring the error as the mean RMS vertex error between the fitting and the test sample. For each test the AAMs were initialized using a center of mass approach by aligning the mean of the model points to the mean of the test data. The starting condition was then corrupted by Gaussian noise with zero mean and a standard deviation of five pixels. The starting condition was perturbed five times per experiment. We performed 70 tests for the ventricle and meat dataset and 180 tests for the face dataset. A total of 250 fittings were performed.

Figure 6 shows the relative performance of the PAAM in comparison to the AAM using ICIA fitting. Figure 5 also shows examples of a fitting for each set. For a suitable selection of the patch-size constant, K , there is a significant amount of overlap between the AAM and the PAAM fitting distributions. This is an encouraging result which shows that the PAAM and the AAM are roughly equivalent models in terms of error when generalizing to new data. Table 1 summarizes the comparatively small texture models used with respect to the AAM and it also shows the difference in CPU² time required to transform each models' texture representation.

The second experiment demonstrated the effect of varying the patch size of the PAAM. By increasing

the patch-size constant K we increase the local texture sizes and potentially introduce overlapping in the texture model. We tested the effect of increasing K from 1 to 20 by performing the previous experiment for each K . The patch size ranged from 4 pixels ($K = 1$) to a maximum of 400 pixels ($K = 20$) and a total of 8400 fittings were performed. Figure 7 shows the effect on the different data sets. For all our data sets the plots show that as K increases, the effect on fitting is relatively small.

The third experiment demonstrated the effect of displacing the starting conditions for the PAAM. This was achieved by generating a “basin of convergence” image. Where each pixel is the mean RMS shape error for a fitting to data outside the training set. We experimented with a translation displacement of -40 to 40 pixels. In the face model case that displacements of up to 20 pixels could be applied (figure 8).

5 Conclusion

This paper has introduced a different patch-transform function for ICIA AAM fitting. This allows for the texture model to be significantly smaller than in the standard approach. Using our patch representation and a suitable selection for patch size, we were able to significantly reduce the number of pixels required for AAM fitting. The reduced texture size translates directly to a reduced fitting time and this is shown in our summary of results. We have also shown that the method, although it only encodes local texture patches, is quite robust with respect to model initialization. Future work will focus on removing the requirement for a patch size constant and automatic initialization.

6 Acknowledgements

The authors would like to thank Georg Langs for providing a corresponding face set and the Australian Research Council for its continued funding.

²This was bench-marked on an Intel Xeon 3.2gigahertz with a MATLAB implementation of the different transforms.

Model Type	Fitting Error (mean)	Fitting Error (std)	Texture Size (pixels)	Transform (seconds)
AAM (face)	4.8	1	30031	0.11
AAM (heart)	1.7	0.3	2261	0.064
AAM (meat)	7.6	2	116748	0.36
PAAM (face,k=5)	4.7	1.3	5838 (19% of AAM)	0.06
PAAM (heart,k=2)	2.3	0.6	1070 (43% of AAM)	0.03
PAAM (meat,k=7)	6.2	2	16295 (13% of AAM)	0.07

Table 1: AAM and PAAM comparison.

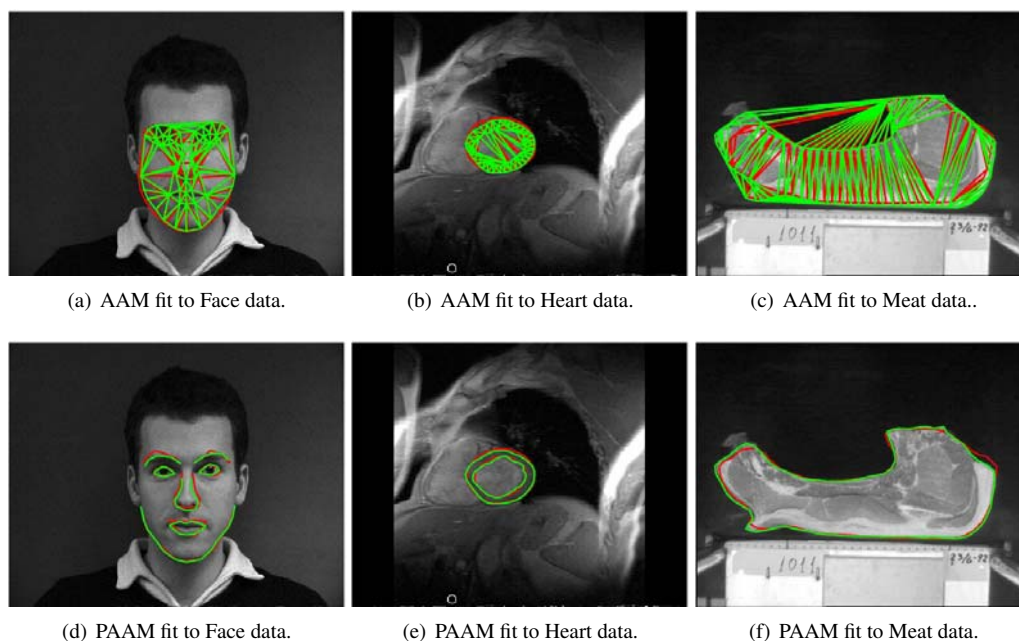
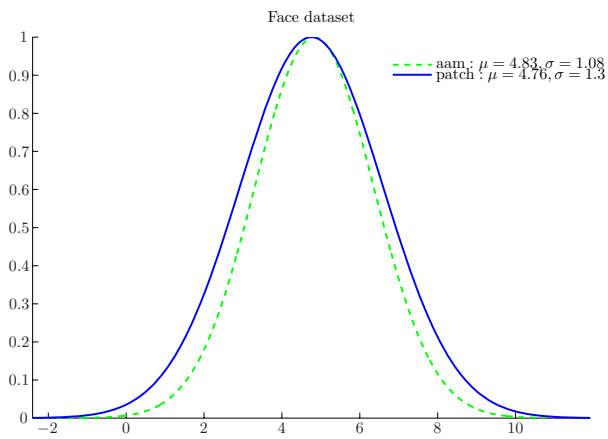


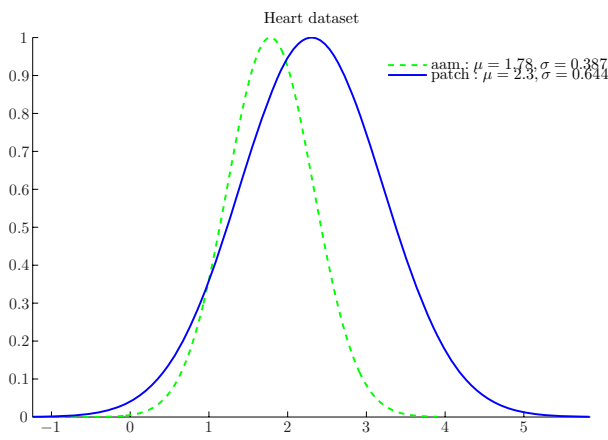
Figure 5: Fitting examples from the test sets, green shape is ground truth, red is AAM/PAAM estimate.

References

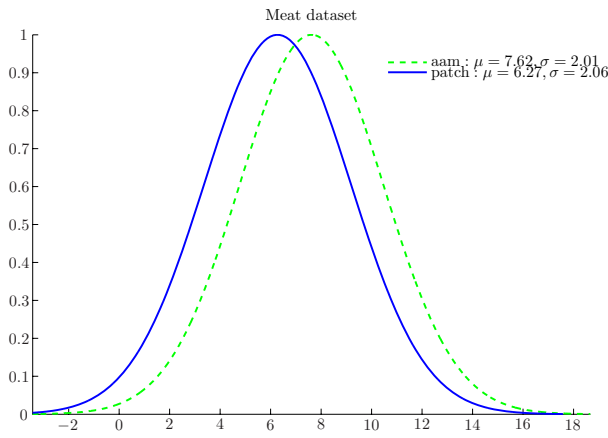
- [1] T. Cootes, G. Edwards, and C. Taylor, "Active appearance models," in *Proc. European Conference on Computer Vision*, vol. 2, pp. 484–498, Springer, 1998.
- [2] M. Devrim, "Generalized Procrustes analysis and its applications in photogrammetry," tech. rep., Swiss Federal Institute Of Technology, 2003.
- [3] C. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
- [4] T. Cootes, C. Taylor, D. H. Cooper, and J. Graham, "Active shape models - their training and application," in *Computer Vision and Image Understanding*, vol. 61, pp. 38–59, 1995.
- [5] I. Matthews and S. Baker, "Active appearance models revisited," *International Journal of Computer Vision*, 2000.
- [6] S. Baker and I. Matthews, "Lucas-Kanade 20 years on; a unifying framework: Part1," Tech. Rep. 16, Robotics Institute, Carnegie Mellon University, 2002.
- [7] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 674–679, 1981.
- [8] R. Gross, I. Matthews, and S. Baker, "Generic vs. person specific active appearance models." The Robotics Institute, Carnegie Mellon University, 2003.
- [9] I. Matthews, J. Xiao, S. Baker, and T. Kanade, "Real-time combined 2D+3D active appearance models," in *Computer Vision and Pattern Recognition*, vol. 2, pp. 535–542, 2004.
- [10] N. Faggian, S. Romdhani, J. Sherrah, and A. Paplinski, "Color active appearance model analysis using a 3D morphable model," in *Digital Image Computing: Techniques and Applications*, December 2005.
- [11] R. Gross, I. Matthews, and S. Baker, "Generic vs. person specific active appearance models," in *British Machine Vision Conference*, September 2004.



(a) Fitting Error on Face data.



(b) Fitting Error on Heart data.

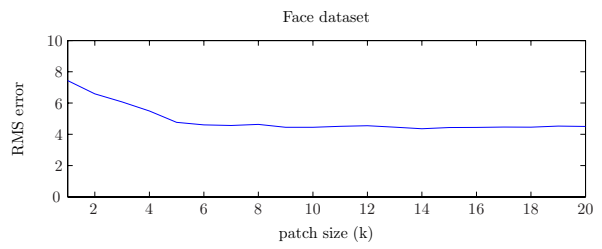


(c) Fitting Error on Meat data.

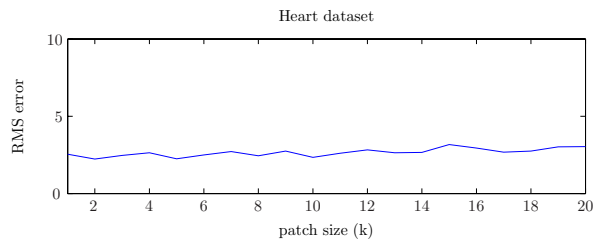
Figure 6: Convergence Properties of the AAM and Patch AAM

[12] M. B. Stegmann, “An annotated dataset of 14 meat images,” tech. rep., Informatics and Mathematical Modelling, Technical University of Denmark, 2002.

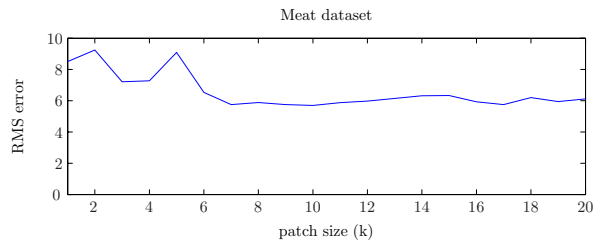
[13] M. B. Stegmann, “An annotated dataset of 14 cardiac MR images,” tech. rep., Informatics and Mathematical Modelling, Technical University of Denmark, 2004.



(a) Face dataset, mean rms fitting error / patch size

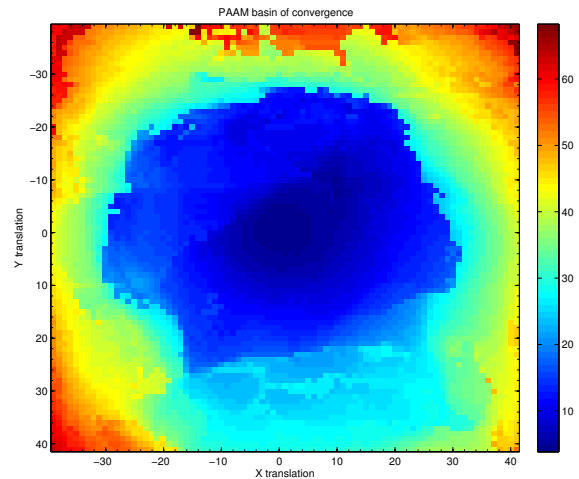


(b) Heart dataset, mean rms fitting error / patch size



(c) Meat dataset, mean rms fitting error / patch size

Figure 7: Effect of varying K for fitting our modified AAM



(a) Face dataset basin of convergence, X and Y model displacements.

Figure 8: PAAM Basin of convergence

Denmark, 2002.

[14] M. M. Nordstrøm, M. Larsen, J. Sierakowski, and M. B. Stegmann, “The IMM face database - an annotated dataset of 240 face images,” tech. rep., Informatics and Mathematical Modelling, Technical University of Denmark, 2004.

Modified Kalman Filtering for Image Super-Resolution

C. Newland, D. Gray and D. Gibbins

Department of Electrical and Electronic Engineering, University of Adelaide, SA 5005, Australia

Email: {cnewland, dgray, danny} @ eleceng.adelaide.edu.au

Abstract

The Kalman filter is usually considered to be too computationally infeasible for image super-resolution. This paper presents a modified Kalman filter super-resolution algorithm, for the case of global translational motion, by approximating the Kalman gain matrix with patterns empirically uncovered within steady state Kalman gain matrices. The resulting algorithm is capable of creating high quality super-resolution images in the order of megapixels, and is tractable for real time implementation.

Keywords: image super-resolution, Kalman filter

1 Introduction

Image super-resolution is the process of estimating a high-resolution image, or image sequence, from a sequence of noisy low-resolution image frames taken of the same scene (or target) but from marginally different perspectives. The differences in the camera position ensure that that the scene has been sampled differently in each image, causing the pixel intensity values to differ between the image frames of the low-resolution sequence. Image super-resolution utilizes these differences to build a higher resolution composite image using all of the information contained within the individual low-resolution frames. Maximum a-posteriori (MAP) and projection onto convex sets (POCS) algorithms were the dominant approaches to image super-resolution during the 1990's [1], with recent research focusing on fast and simple algorithms aiming for real-time applications [2, 3].

Kalman filters are widely acknowledged for creating the optimal mean square error estimate in the context of linear constraints. However, in the field of image super-resolution they are equally recognised as being computationally unfeasible because the update of the Kalman gain matrix requires a large matrix inversion. Previously published approximations to the Kalman filter have included: recursive steepest-descent (R-SD) and recursive least squares (R-LMS) algorithms that avoid the matrix inversion by approximating the Kalman filter itself [4, 5]; reduced update Kalman filters that approximate the state-space into smaller regions to minimise the processing required for the matrix inversion [6]; and an approximation for global translational motion with constant blur formed by removing the blurring matrix from the Kalman filter formulation [3].

This paper presents a novel simplified time varying Kalman filter for image super-resolution for the special case of global translational motion. Empirical patterns uncovered by the steady-state Kalman gain matrices have been used to approximate the time varying Kalman gain matrices, and significantly improve the computational complexity of the Kalman filter with minimal loss of information.

Section 2 of this paper provides background details on Kalman filtering. In section 3, the modified Kalman filter algorithm will be formulated. Results from trials on both simulated and real imagery are presented in section 4, with a summary of major findings provided in section 5.

2 Background

2.1 Problem Statement

The original scene may be approximated by a high-resolution image that provides an improvement in resolution by the magnification factor, m , relative to the low-resolution pixels measurements. Between each frame in the sequence, the imaging system experiences a globally translational motion relative to the scene of interest. To simplify processing, the low and high-resolution images are converted into 1-dimensional vectors using column-wise lexicographical ordering. The vector containing the pixel values from the k^{th} low-resolution frame is denoted as \mathbf{y}_k , with the underlying high-resolution pixels contained in \mathbf{x}_k .

The relationship between the low-resolution frames and the high-resolution image is given by equation (1). The measurement matrix, \mathbf{H}_k , models the blur and decimation of the camera sensor used to create the low-resolution frame \mathbf{y}_k . Errors in

this modelling, as well as any further system aberrations and noise, are modelled as Gaussian measurement noise, \mathbf{v}_k , with covariance matrix $\mathbf{R}_k = \sigma_{R_k}^2 \mathbf{I}$ and zero mean.

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \quad (1)$$

The motion of the sensor with respect to the scene of interest is assumed to be known and is modelled by a system matrix, \mathbf{F}_k . Possible errors in the motion model are considered to be Gaussian system noise, \mathbf{w}_k , with covariance matrix $\mathbf{Q}_k = \sigma_{Q_k}^2 \mathbf{I}$ and zero mean. The change in location of the sensor's field of view is modelled by equation (2), where \mathbf{x}_k corresponds to the current high-resolution image and \mathbf{x}_{k+1} corresponds to the high-resolution image after the next camera motion when \mathbf{y}_{k+1} enters the field of view.

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{w}_k \quad (2)$$

2.2 Kalman Filtering

The states of the Kalman filter, $\hat{\mathbf{x}}_{k|k}$, are the super-resolution estimates of the underlying high-resolution scene \mathbf{x}_k . Every time increment, k , of the Kalman filter will correspond to the sequential input of a low-resolution image frame, \mathbf{y}_k .

The Kalman filter prediction equation in (3) takes the current super-resolution image after processing frame $k - 1$, $\hat{\mathbf{x}}_{k-1|k-1}$, and applies the transition matrix, \mathbf{F}_k , to predict the next super-resolution estimate, $\hat{\mathbf{x}}_{k|k-1}$. The correction equation in (4) updates the prediction from equation (3) with a weighted error signal. The error signal is the difference in low-resolution pixel values between the next frame, \mathbf{y}_k , and those that would have been created from the predicted super-resolution image when passed through the measurement matrix, \mathbf{H}_k . The Kalman gain matrix, \mathbf{K}_k calculated using equation (5), applies and weights the low-resolution pixel error signal to correct the high-resolution pixels of the predicted super-resolution estimate.

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} \quad (3)$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}) \quad (4)$$

$$\mathbf{K}_k = \Sigma_{k|k-1} \mathbf{H}_k^T (\mathbf{H}_k \Sigma_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (5)$$

The error covariance matrices, $\Sigma_{k|k}$ and $\Sigma_{k+1|k}$, are also calculated in a similar prediction and correction process as shown in equations (6) and (7).

$$\Sigma_{k+1|k} = \mathbf{F}_k \Sigma_{k|k} \mathbf{F}_k^T + \mathbf{Q}_k \quad (6)$$

$$\Sigma_{k|k} = \Sigma_{k|k-1} - \mathbf{K}_k \mathbf{H}_k \Sigma_{k|k-1} \quad (7)$$

Every update of the Kalman filter requires the inversion of a matrix formed by $(\mathbf{H}_k \Sigma_{k|k} \mathbf{H}_k^T + \mathbf{R}_k)$,

with each of the square dimensions equal to the total number of pixels in a low-resolution frame. This inversion requires an unfeasible amount of processing for standard sized images, and it has been the avoidance of this inversion that has led to approximations of the Kalman filter in past literature [4, 5, 6, 3] and within this paper.

2.3 Steady state Kalman filtering

Steady state is considered to occur when neither the motion between frames, \mathbf{F}_k , nor the point spread function of the camera sensor, \mathbf{H}_k , changes between the frames of the low-resolution sequence. The error covariance matrix then becomes equal to the solution of the Algebraic Ricatti Equation (ARE) given in equation (8), and the Kalman gain matrix needs only to be calculated once using equation (9).

$$\Sigma = \mathbf{F} \Sigma \mathbf{F}^T - \mathbf{F} \Sigma \mathbf{H}^T (\mathbf{H} \Sigma \mathbf{H}^T + \mathbf{R})^{-1} \mathbf{H} \Sigma \mathbf{F}^T + \mathbf{Q} \quad (8)$$

$$\mathbf{K} = \Sigma \mathbf{H}^T (\mathbf{H} \Sigma \mathbf{H}^T + \mathbf{R})^{-1} \quad (9)$$

3 Modified Kalman Filter

The aim of the Kalman filter modification presented here, is to utilise patterns from the steady-state Kalman gain matrix to form an approximate reconstruction of time varying Kalman gain matrices. The development, formulation and computational complexity is covered below.

3.1 Modified Kalman Filter Development

The steady state Kalman gain matrix produced by equation (9) is a relatively sparse matrix with a row dimension equal to the total number of high-resolution pixels and a column dimension equal to the total number of low-resolution pixels in a frame. Most of the matrix elements are significantly small in magnitude compared to the main positive peaks observed in the matrix. An example row from a Kalman gain matrix is plotted in figure 1. By applying a threshold to the Kalman gain matrix, the positive peaks can be extracted. Notably, the structure of this reduced-element matrix matches that of the transpose of the measurement matrix, irrespective of the choice of blur and decimation operations. Example structure (spy) plots of \mathbf{H}^T and the resulting reduced element Kalman gain matrix are shown in figure 2 for both a simple spatial averaging and decimation operation and a random measurement matrix. For the remainder of this paper, it has been assumed that the sensor point spread function can be approximated by

a simple square “top-hat” spatial averaging and decimation operation.

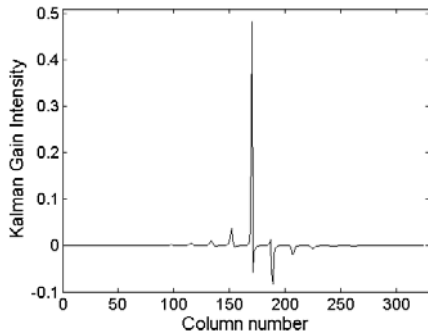


Figure 1: Example row of steady state Kalman gain matrix [$m=2$, $QR=1$, & (1,1) motion].

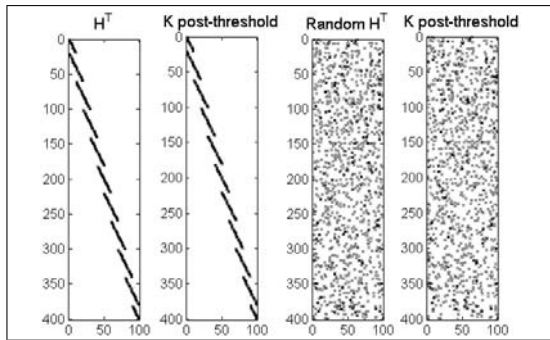


Figure 2: H^T and reduced-element steady state K matrix structures for $m=2$ (left) and random blur (right).

As each row of the Kalman gain matrix corresponds with one high-resolution pixel, and each column corresponds to one low-resolution pixel, the largest positive peaks within the matrix lie where the low-resolution pixels contain their respective high-resolution pixels. The other minor peaks, as shown in figure 1, correspond to small error contributions at the edges of neighbouring low-resolution pixels. Examination of these relationships uncovered two results that apply without discrimination: at some locations these minor peaks enhance the edges of the low-resolution pixel areas; but at other locations the minor peaks enhance one side of the low-resolution pixel edge and smooth the other side. Given that the Kalman gain matrix is formed independently from the image sequence, the contrast in these two functions suggest that the minor peaks are likely to be residual effects formed in the process of solving the ARE.

When the positive peaks are extracted and displayed in a 2-dimensional format according to their corresponding high-resolution pixel position, repeated patterns become apparent as shown in figure 3. Each of the repeating pattern blocks corresponds to a low-resolution pixel, with the

internal pattern of that block indicating the weightings given to the individual high-resolution pixels. The internal patterns were found to change with different sensor motions and system-to-measurement noise variance ratios ($QR=\sigma_Q^2/\sigma_R^2$). Figure 3 displays a progression of the gain patterns for an example case of a magnification by 4 and a constant one pixel diagonally down and right sensor motion. While the leading edges within this figure have low gain values, the patterned sections are relatively constant.

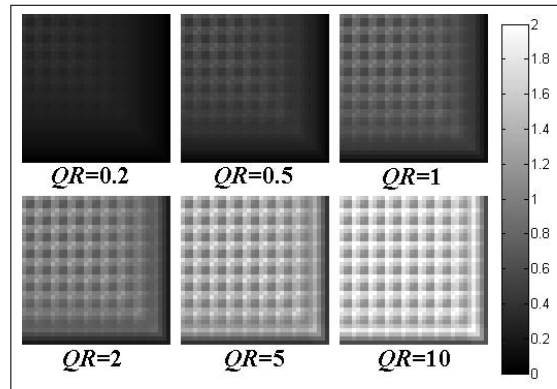


Figure 3: Reduced-element gain patterns with increasing QR . [$m=4$, (1,1) motion]

Further investigations showed that the individual high-resolution pixel gain weightings within each low-resolution pixel pattern were found to have logarithmic relationships with the QR ratio, as shown in figure 4. By disregarding the QR ratios below 0.2 where the measurements are too noisy to be useful, and the QR ratios above 10 where the motion estimation is too inaccurate to be useful, the central portion of the logarithmic QR ratios plots can be approximated by a linear relationship. These linear relationships form the foundation of the modified Kalman filter formulation presented here.

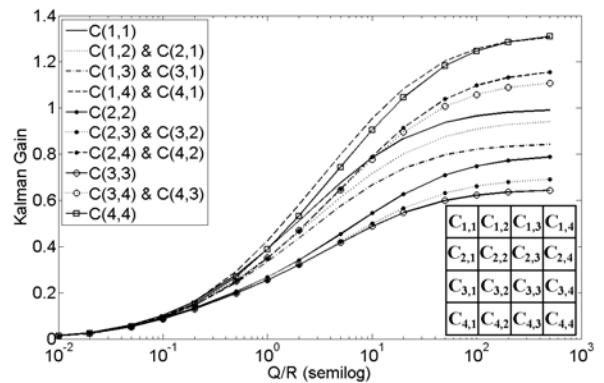


Figure 4: Logarithmic relationship between reduced element gains and corresponding pixel locations. [$m=4$, (1,1) motion]

3.2 Modified Kalman Filter formulation

The modified Kalman filter presented in this paper creates an approximate Kalman gain matrix, $\hat{\mathbf{K}}_k$, based on the reduced-element Kalman gain matrix discussed in the previous section. As given in equation 11, $\hat{\mathbf{K}}_k$ is created by multiplying the structure of the transpose of the measurement matrix by a diagonal matrix, \mathbf{D}_k , based on the subpixel motion of frame k . The diagonal elements of \mathbf{D}_k are the approximations of the patterned high-resolution pixel gain weightings discussed in section 3.1, rearranged back into vector notation.

$$\hat{\mathbf{K}}_k = \mathbf{D}_k \cdot \text{structure}(\mathbf{H}^T) \quad (10)$$

By assuming a constant 2-dimensional pattern in the positive peaks of the Kalman gain matrix, the high-resolution pixel gain weightings within each low-resolution block pattern may be defined as per figure 5. For any given QR ratio, and subpixel motion between frames, the approximate high-resolution gain weightings within \mathbf{D}_k may then be calculated in the form of equation (11). Table 1 contains a collection of the empirically derived coefficients for these relationships with magnifications of 2, 3 and 4.

$$A_{1,1} = \text{Slope}(A_{1,1}) * \log_{10}(QR) + \text{Const}(A_{1,1}) \quad (11)$$

$A_{1,1}$	$A_{1,2}$	$B_{1,1}$	$B_{1,2}$	$B_{1,3}$	$C_{1,1}$	$C_{1,2}$	$C_{1,3}$	$C_{1,4}$
		$B_{2,1}$	$B_{2,2}$	$B_{2,3}$	$C_{2,1}$	$C_{2,2}$	$C_{2,3}$	$C_{2,4}$
$A_{2,1}$	$A_{2,2}$	$B_{3,1}$	$B_{3,2}$	$B_{3,3}$	$C_{3,1}$	$C_{3,2}$	$C_{3,3}$	$C_{3,4}$
					$C_{4,1}$	$C_{4,2}$	$C_{4,3}$	$C_{4,4}$

Figure 5: High-resolution pixel gain weighting labels for $m=2$ (A's), 3 (B's) & 4 (C's).

The subpixel motions of the camera sensor can be defined as z high-resolution columns to the right and y high-resolution rows downwards. Larger sensor motions can be broken into two parts: low-resolution pixel shifts that are applied directly to the super-resolution estimate; and subpixel shifts. It should be noted that, through symmetry, the full range of positive and negative subpixel motions are not required to be considered independently:

- If $z < 0$ (leftward camera motion) then consider $z = \text{abs}(z)$ and horizontally flip each low-resolution pixel block of gain values.
- If $y < 0$ (upwards camera motion) then consider $y = \text{abs}(y)$ and vertically flip each low-resolution pixel block of gain values.
- If $y > z$ then swap the z and y values, and take the transpose of each low-resolution pixel block of gain values.

- If $z = 0$ and $y = 0$ (stationary motion) choose to set the gain to be equal to the lowest gain equation for that magnification. Note that it is not possible to solve the ARE directly for the case of stationary motion as the eigenvalues of the error covariance matrix fall too close to the unit circle.

For a magnification of 2, the coefficients within table 1 are essentially the same irrespective of the gain positioning or the motion between frames. A time-invariant approximation of \mathbf{D}_k can therefore be made by calculating a single value, such as $0.35 \log_{10}(QR) + 0.48$, to apply to the entire structure of the \mathbf{H}^T matrix. This provides a slightly faster modified Kalman filter algorithm option with minimal loss of information.

The complete modified Kalman filter formulation can be used to form super-resolution images, or image sequences, over any length of image frames or alternatively as a sliding window algorithm. Further enhancements can easily be incorporated by changing the QR ratio between frames, or within the individual low-resolution pixel blocks as desired.

3.3 Computational Complexity

For a $p \times q$ pixel low resolution frame with a desired magnification of m , every update of the modified Kalman filter will require: A shifting operation on the super-resolution estimate; $2m^2pq$ element additions; $(m^2+1)pq$ element multiplications; pq element subtractions; and $2m^2$ lookups. This computationally efficient algorithm is capable of creating super-resolution images in the order of megapixels, and is attractive for real time implementation.

3.4 Related Methods

In [3], the Kalman filter for global translational motion utilised the assumption that \mathbf{H} (excluding the decimation operation) and \mathbf{F}_k were block circulant and commutable. While this is true for the \mathbf{H} matrix, the \mathbf{F}_k matrix will only be fully block circulant if the pixels leaving the field of view are wrapped to the opposite edge of the super-resolution estimate. Despite this assumption, the empirical results agree with [3] in that the \mathbf{H} matrix is extractable from the Kalman filtering process, leaving only a diagonal gain matrix. The matrix formed within this paper relates to the high-resolution pixels formed from a spatial averaging process whereas [3] creates a matrix of low-resolution relationships, and estimates the high-resolution pixels as a separate bilateral filtering process.

Table 1: Coefficients of equation (11) for magnifications of 2, 3 & 4 and various subpixel motions.

Magnification = 2				Magnification = 4				Magnification = 4			
Motion (z,y)	Pos.	Slope	Const.	Motion (z,y)	Pos.	Slope	Const.	Motion (z,y)	Pos.	Slope	Const.
(0,0)	$A_{1-2,1-2}$	0.3457	0.4703	(0,0)	$C_{1-4,1-4}$	0.1902	0.2370	(3,1)	$C_{1,1}$	0.5066	0.3661
(1,0)	$A_{1-2,1}$	0.3578	0.4810	(1,0)	$C_{1-4,1}$	0.4044	0.3969		$C_{2,1}$	0.4371	0.3346
	$A_{1-2,2}$	0.3457	0.4703		$C_{1-4,2}$	0.2372	0.2609		$C_{3,1}$	0.4331	0.3333
(1,1)	$A_{1,1}$	0.3527	0.4963		$C_{1-4,3}$	0.1902	0.2370		$C_{4,1}$	0.4875	0.3564
	$A_{1,2}$ & $A_{2,1}$	0.3529	0.4944		$C_{1-4,4}$	0.4684	0.4135		$C_{1,2}$	0.3443	0.2953
	$A_{2,2}$	0.3522	0.4935	(2,0)	$C_{1-4,1-2}$	0.3139	0.2892		$C_{2,2}$	0.2748	0.2638
Magnification = 3					$C_{1-4,3-4}$	0.3004	0.2751		$C_{3,2}$	0.2708	0.2625
Motion (z,y)	Pos.	Slope	Const.						$C_{4,2}$	0.4276	0.3312
(0,0)	$B_{1-3,1-3}$	0.2263	0.2916	(3,0)	$C_{1-4,1}$	0.4965	0.3488		$C_{1,3}$	0.3607	0.3010
(1,0)	$B_{1-3,1}$	0.3754	0.4432		$C_{1-4,2}$	0.2510	0.2457		$C_{2,3}$	0.2911	0.2695
	$B_{1-3,2}$	0.2345	0.2858		$C_{1-4,3}$	0.2738	0.2535		$C_{3,3}$	0.2707	0.2624
	$B_{1-3,3}$	0.4032	0.4522		$C_{1-4,4}$	0.3572	0.2741		$C_{4,3}$	0.4275	0.3312
(2,0)	$B_{1,1-3}$	0.4406	0.4525	(1,1)	$C_{1,1}$	0.3752	0.4044		$C_{1,4}$	0.3621	0.2839
	$B_{2,1-3}$	0.2263	0.2916		$C_{2,1}$ & $C_{1,2}$	0.3393	0.3685		$C_{2,4}$	0.3448	0.2765
	$B_{3,1-3}$	0.3738	0.4085		$C_{3,1}$ & $C_{1,3}$	0.3137	0.3479		$C_{3,4}$	0.3279	0.2708
(1,1)	$B_{1,1}$	0.3510	0.4459		$C_{4,1}$ & $C_{1,4}$	0.4768	0.4550		$C_{4,4}$	0.4630	0.3293
	$B_{2,1}$ & $B_{1,2}$	0.3132	0.3973		$C_{2,2}$	0.2432	0.2837	(3,2)	$C_{1-2,1}$	0.4423	0.3356
	$B_{3,1}$ & $B_{1,3}$	0.4006	0.4833		$C_{3,2}$ & $C_{2,3}$	0.2193	0.2667		$C_{3-4,1}$	0.4420	0.3355
	$B_{2,2}$	0.2543	0.3213		$C_{4,2}$ & $C_{2,4}$	0.3840	0.3775		$C_{1-2,2}$	0.3330	0.2882
	$B_{3,2}$ & $B_{2,3}$	0.3410	0.4085		$C_{3,3}$	0.2132	0.2660		$C_{3-4,2}$	0.3328	0.2882
	$B_{3,3}$	0.3944	0.4630		$C_{3,4}$ & $C_{4,3}$	0.3782	0.3767		$C_{1-2,3}$	0.3383	0.2908
(2,1)	$B_{1,1}$	0.4201	0.4757		$C_{4,4}$	0.4512	0.4231		$C_{3-4,3}$	0.3381	0.2907
	$B_{2,1}$	0.3507	0.4107	(2,1)	$C_{1,1-2}$	0.3883	0.3482		$C_{1-2,4}$	0.3656	0.2855
	$B_{3,1}$	0.4119	0.4588		$C_{2,1-2}$	0.3290	0.3146		$C_{3-4,4}$	0.3655	0.2854
	$B_{1,2}$	0.3219	0.3859		$C_{3,1-2}$	0.3127	0.3084	(3,3)	$C_{1,1}$	0.4939	0.3594
	$B_{2,2}$	0.2528	0.3240		$C_{4,1-2}$	0.4311	0.3694		$C_{2,1}$ & $C_{1,2}$	0.4332	0.3340
	$B_{3,2}$	0.3511	0.4033		$C_{1,3-4}$	0.3727	0.3289		$C_{3,1}$ & $C_{1,3}$	0.4343	0.3344
	$B_{1,3}$	0.3665	0.4229		$C_{2,3-4}$	0.3142	0.2977		$C_{4,1}$ & $C_{1,4}$	0.4714	0.3331
	$B_{2,3}$	0.3200	0.3841		$C_{3,3-4}$	0.2993	0.2920		$C_{2,2}$	0.2729	0.2646
	$B_{3,3}$	0.4183	0.4634		$C_{4,3-4}$	0.4184	0.3509		$C_{3,2}$ & $C_{2,3}$	0.2740	0.2650
(2,2)	$B_{1,1}$	0.4133	0.4616	(2,2)	$C_{1-2,1-2}$	0.3229	0.3051		$C_{4,2}$ & $C_{2,4}$	0.3335	0.2740
	$B_{2,1}$ & $B_{1,2}$	0.3525	0.4056		$C_{3-4,1-2}$	0.3203	0.2932		$C_{3,3}$	0.2940	0.2721
	$B_{3,1}$ & $B_{1,3}$	0.4208	0.4664		$C_{1-2,3-4}$	0.3203	0.2932		$C_{4,3}$ & $C_{3,4}$	0.3500	0.2798
	$B_{2,2}$	0.2517	0.3242		$C_{3-4,3-4}$	0.3178	0.2920		$C_{4,4}$	0.3684	0.2876
	$B_{3,2}$ & $B_{2,3}$	0.3200	0.3850								
	$B_{3,3}$	0.3662	0.4237								

Essentially the algorithm presented in this paper has the same processing requirements as the “simulate and correct” algorithms of the late 1980’s [1] but with the advantages of the Kalman filter.

4 Results

4.1 Simulated data

Simulated sequences of low-resolution frames were created from model high-resolution images by performing square “top hat” spatial averaging and decimation on shifted cropped regions. Example imagery is shown in figure 6 for magnifications of 4. With perfect registration and matched point spread functions, very good reconstructions were achieved but mild halo effects occasionally became apparent along severe black/white edges. Future work will be aimed at correcting this ringing, and investigating the effects of noise and motion estimation errors.

4.2 Real data

The modified Kalman filter algorithm has been applied to real data in the form of video sequences ob-

tained from unmanned aerial vehicle (UAV) flights and image sequences from confocal microscopes. Neither the camera point spread functions, nor the motion between frames were known, providing an opportunity to examine how well the “top hat” point spread function approximation can be applied to real imagery. Promising results have been obtained to date, with example imagery shown in figure 7. In these examples, simple multi-scale correlation was used to align the image frames. As per most super-resolution algorithms, the quality of the reconstruction using this paper’s algorithm is highly dependent on the quality of the motion estimation.

5 Conclusion

This paper has proposed a modified Kalman filter algorithm for image super-resolution, that is based on empirical patterns uncovered in the steady state Kalman gain matrices. Good super-resolution reconstructions have been achieved with both simulated and real imagery, despite the assumption of an ideal “top hat” spatial averaging point spread function. The algorithm is computationally effi-

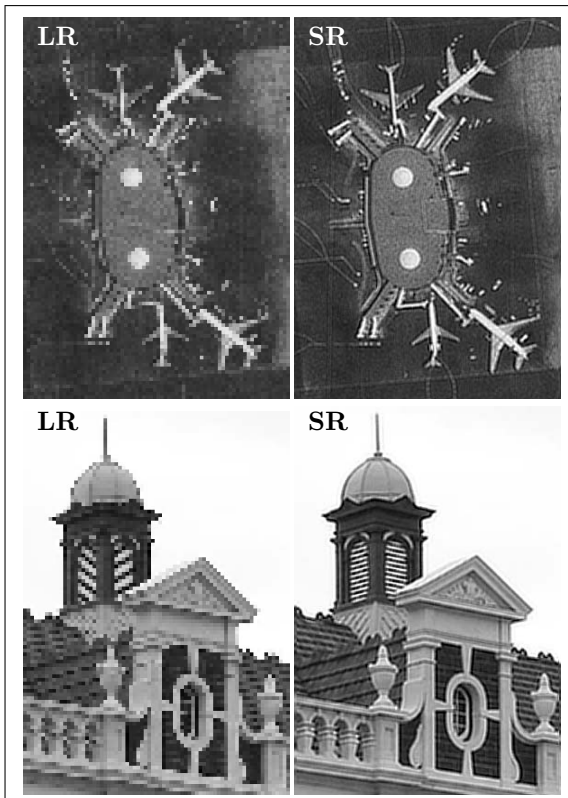


Figure 6: Super-resolution images ('SR') formed from simulated low-resolution imagery ('LR').

cient as compared to a full Kalman filter, allows super-resolution images in the order of megapixels to be created, and is attractive to real time implementation. Future investigations are aimed at comparing this algorithm more closely to current techniques, reducing the halos apparent at severe black/white edges, expanding the algorithm to colour super-resolution and focussing on specific applications.

6 Acknowledgements

We acknowledge the support of DSTO and Adelaide Microscopy in providing the imagery used in section 4.2 of this paper.

References

- [1] S. Borman and R. Stevenson, "Spatial resolution enhancement of low-resolution image sequences a comprehensive review with directions for future research," tech. rep., Laboratory for Image and Signal Analysis (LISA), University of Notre Dame, 8 July 1998.
- [2] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, "Advances and challenges in super-resolution," *International Journal of Imaging Systems and Technology, Special Issue on High*

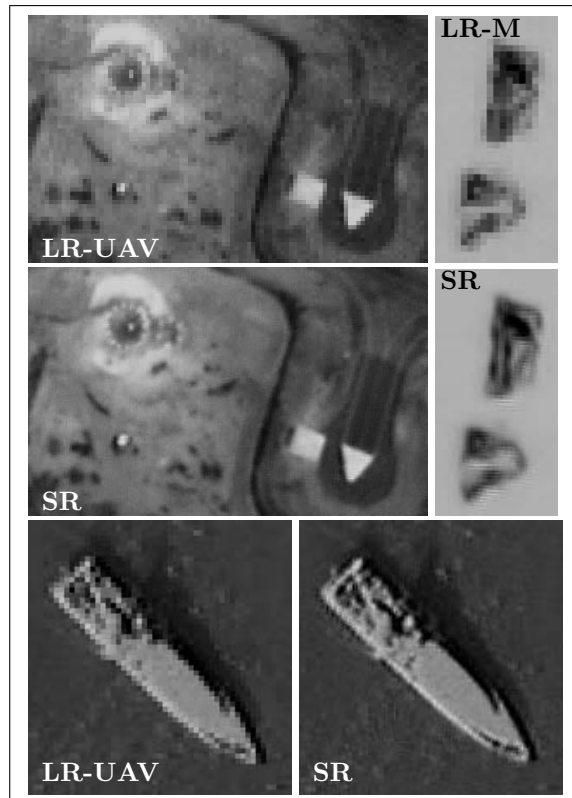


Figure 7: Super-resolution images ('SR') formed from low-resolution UAV video sequences ('LR-UAV') and confocal microscopy ('LR-M') imagery.

Resolution Image Reconstruction, vol. 14, no. 2, pp. 47–57, 2004.

- [3] S. Farsiu, M. Elad, and P. Milanfar, "Video-to-video dynamic superresolution for grayscale and color sequences," *EURASIP Journal of Applied Signal Processing, Special Issue on Superresolution Imaging*, pp. 1–15, 2006.
- [4] M. Elad and A. Feuer, "Super-resolution reconstruction of continuous image sequences," in *1999 International Conference on Image Processing (ICIP 99), Proceedings of the*, vol. 3, (Kobe, Japan), pp. 459–463, 1999.
- [5] M. Elad and A. Feuer, "Superresolution restoration of an image sequence: adaptive filtering approach," *Image Processing, IEEE Transactions on*, vol. 8, no. 3, pp. 387–395, 1999.
- [6] A. Patti, A. Tekalp, and M. Sezan, "A new motion-compensated reduced-order model kalman filter for space-varying restoration of progressive and interlaced video," *Image Processing, IEEE Transactions on*, vol. 7, no. 4, pp. 543–554, 1998.

Affine Normalized Contour Invariants using Independent Component Analysis and Dyadic Wavelet Transform

Asad Ali S.A.M Gilani

Faculty of Computer Science and Engineering
Ghulam Ishaq Khan Institute of Engineering Sciences and Technology
Topi-23460, Swabi, NWFP
Pakistan
Email: { aali, asif }@giki.edu.pk

Abstract

The paper presents a hybrid technique for affine invariant feature extraction with the view of object recognition based on parameterized contour. The proposed technique first normalizes an input image by removing affine distortions using independent component analysis which also reduces the effect of noise introduced during contour parameterization. Then two invariant functionals at three different dyadic levels are constructed using the wavelet based conic equation. Experimental results conducted using three different standard datasets confirm the validity of the proposed approach. Beside this the error rates obtained in terms of invariant stability are significantly lower when compared to other wavelet based invariants and the proposed invariants exhibit higher feature disparity than the method of Fourier descriptors.

Keywords: Affine invariants, Independent Component analysis, Dyadic Wavelet Transform, Conics, Geometric Transformations, Pattern recognition.

1 Introduction

One of the key tasks in robotic vision is to recognize objects when subjected to different viewpoint transformations and this can be achieved by constructing invariants to certain groups (Euclidean, affine, projective transformations) which hold potential for widespread applications for industrial part recognition [14], handwritten character recognition [15], identification of aircrafts [6], and shape analysis [16] to name a few. Viewpoint related changes of objects can broadly be represented by weak perspective transformation which occurs when the depth of an object along the line of sight is small compared to the viewing distance. This reduces the problem of perspective transformation to the affine transformation which is linear [18].

The affine group includes the four basic forms of geometric distortions, under weak perspective projection assumption, namely translation rotation, scaling and shearing. Finding a set of descriptors that can resist geometric attacks on the object contour can act as a good starting point for the more difficult projective group of transformations.

In this paper we propose a new method of constructing invariants which is based on normalizing an affine distorted and noise corrupted object boundary using independent component analysis which makes it invariant to translation, scaling and shearing deformations beside removing noise from the contour data points. Then using the restored object contour we construct two invariants using the approximation coefficients of the dyadic wavelet

transform. It is important to mention here that the constructed invariants are independent of the contour scan order.

The rest of the paper is organized as follows. In section 2 we review some the previously published works, section 3 describes the proposed method in detail and section 4 provides experimental results and comparisons with previously published techniques. Let us have a brief overview of independent component analysis before going into the details.

1.1 Independent Component Analysis

Primarily developed to find a suitable representation of multivariate data it performs blind source separation of a linear mixture of signals and has found numerous applications in short time. Assume that we observe a linear mixture Q of n independent components:

$$Q_j = A_{j1}S_1 + A_{j2}S_2 + \dots + A_{jn}S_n \text{ for all } j \quad (1)$$

where A represents the mixing variable and S the source signals. Using vector notation it can be expressed as:

$$Q = AS \quad (2)$$

The model above is called the independent component analysis or ICA model [1][2] which is a generative model as it describes the process of mixing the component signals S_i . All that is observed is Q and A , S must be estimated from it. In order to estimate A , the component S_i must be statistically independent and have a non-gaussian distribution. After estimating the mixing variable A we can compute its inverse say W and obtain the independent components as:

$$S = WQ \quad (3)$$

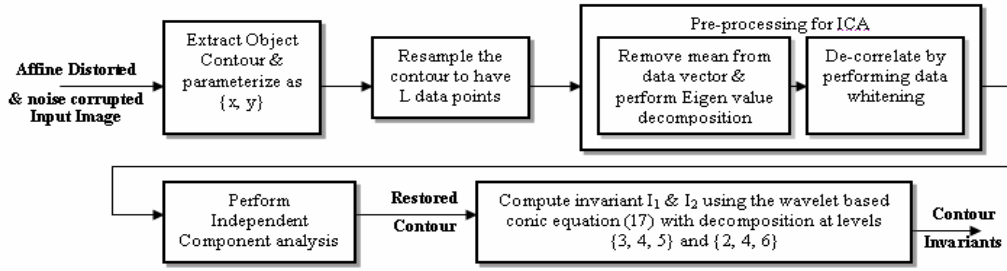


Figure 1 shows the complete system diagram for the construction of contour based invariants.

We opted for ICA as a possible solution space because an affine deformation of the object contour results in the linear mixing of the data points on the coordinate axis besides being coupled with random noise during contour parameterization.

2 Related Work

Keeping in view the importance of constructing invariants and their widespread applications research has been conducted by many which can broadly be classified into two groups namely: Region based and Contour based invariant descriptors. In the context below we review some of the contour based techniques that are most related to the present work.

Several parameterizations of the object boundary that are linear under an affine transformation have been proposed. The affine arc length τ proposed in [8] is defined as:

$$\tau = \int_a^b \sqrt[3]{x(t)'y(t)'' - x(t)''y(t)'} dt \quad (4)$$

where $x(t)'$, $y(t)'$ are the first and $x(t)''$, $y(t)''$ the second order derivatives with respect to the parameterization order t . As the above computation requires second order derivatives so it becomes susceptible to noise introduced because of incorrect segmentation of the object.

To solve the above problem Arbter *et al.* [9] introduced the invariant Fourier descriptors using the enclosed area parameter defined as:

$$\sigma = \frac{1}{2} \int_a^b |x(t)y(t)' - y(t)x(t)'| dt \quad (5)$$

The above formulation was derived using the property that the area occupied by an object changes linearly under an affine transformation. The only drawback is that it is not invariant to translation and requires the starting and ending points to be connected. Arbter also found that using sign in the enclosed area parameter (5) makes it much less sensitive to noise instead of the absolute values. Beside this the technique has a higher misclassification rate as compared to the wavelet based descriptors.

Zhao *et al.* [10] introduced affine curve moment invariants based on affine arc length (4) defined as:

$$v_{pq} = \int_C [x(t) - \tilde{x}]^p [y(t) - \tilde{y}]^q \{ [x(t) - \tilde{x}]y(t)' [y(t) - \tilde{y}]x(t)' \} dt \quad (6)$$

where \tilde{x} and \tilde{y} are the centroid of the contour computed using (4) after removing the cubic root in the framework of moments. They derived a total of three invariants using equation (6) and have shown them to be invariant to the affine group of transformations. The drawback of the above framework is that the invariants are sensitive to noise and local variations of shape because the computation of invariants is based on moments and derivatives of first order.

More recently Manay *et al.* [7] introduced the Euclidean integral invariants to counter the effect of noise based on the concept of differential invariants. They have derived two invariants namely; distance integral invariant and area integral invariant. The major drawback of their work is that the distance integral invariant is a global descriptor and a local change of shape i.e. missing parts of shape, effects the invariant values for the entire shape, whereas the area integral invariant only counters for the Euclidean group of transformations.

Tieng *et al.* [4] proposed the use of dyadic wavelet transform for constructing invariants using the approximation and detail coefficients. They formulated a framework based on enclosed area parameter for constructing invariants in the wavelet domain. Later Khalil *et al.* [5][6] extended their work and derived invariants using the detail coefficients and wavelet based conic equation.

More recently Ibrahim *et al.* [3] derived invariants using the approximation coefficients based on the framework proposed in [4] and showed that approximation based invariants outperform detail based invariants in terms of error rates. We make use of the framework proposed in [5][6] while constructing invariants in the next section and improve upon the wavelet based methods by reducing error rates.

In short we improve on many of the shortcomings mentioned previously.

3 Proposed Technique

We propose a three step process for the construction of contour based invariant descriptors of the objects. The first step acts as foundation for second and third steps in which ICA is applied and then invariants are constructed. Next we provide the detailed description of each step:

3.1 Boundary Parameterization and Re-sampling

In the first step object contour is extracted and parameterized. Let us define this parametric curve as $[x(t), y(t)]$ with parameter t on a plane. Next the parameterized boundary is resampled to a total of L data points. Thus a point on the resampled curve under and affine transformation can be expressed as:

$$\begin{aligned}\tilde{x}(t) &= a_0 + a_1x(t) + a_2y(t) \\ \tilde{y}(t) &= b_0 + b_1x(t) + b_2y(t)\end{aligned}\quad (7)$$

The above equations can be written in matrix form as:

$$\begin{aligned}\begin{bmatrix} \tilde{x}(t') \\ \tilde{y}(t') \end{bmatrix} &= \begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix} \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} + \begin{bmatrix} a_0 \\ b_0 \end{bmatrix} \\ \begin{bmatrix} \tilde{x}(t') \\ \tilde{y}(t') \end{bmatrix} &= P \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} + B \\ Y'(t') &= PY(t) + B\end{aligned}\quad (8)$$

where t and t' are different because of the difference in contour scan order and sampling of the two contours, and Y' is obtained as a result of linear affine transformation of Y , P is the affine transformation matrix and B is the translation vector which can be removed ($B = 0$) by using the centroid contour coordinates.

3.2 Theoretical Formulation and application of ICA

We know that $Y(t)$ and $Y'(t')$ are the linear combination of the same source S with a different mixing matrix A and A' referring to equation (2). Then we can write:

$$\begin{aligned}Y(t) &= AS(t) \\ Y'(t') &= A'S(t)\end{aligned}\quad (9)$$

where A' is the linear combination of P and random noise N . In (9) the mixing matrix A' is different because of the difference in affine transformation parameters and the random noise introduced during contour parameterization.

Next we estimate the mixing variable A' by finding a matrix W of weights using the Fast ICA algorithm from [1]. Then W will be used to find the original source S as per equation (3). The two step process for computing ICA is as follows:

Step 1: Whiten the Centered Data

Whitening is performed on $Y'(t')$ in order to reduce the number of parameters that need to be estimated. Its utility resides in the fact that the new mixing

matrix \tilde{A}' that will be estimated is orthogonal such that it satisfies:

$$\tilde{A}' \tilde{A}'^T = I \quad (10)$$

So, the data Y' becomes uncorrelated after this step. Whitening is then performed by computing the Eigen value decomposition of covariance matrix as:

$$\begin{aligned}Y' Y'^T &= EDE^T \\ \tilde{Y}' &= ED^{-1/2}E^T Y'\end{aligned}\quad (11)$$

where E is the orthogonal matrix of eigenvectors of $\{Y' Y'^T\}$ and D is the diagonal matrix of eigen values.

Step 2: Apply ICA on the Whitened Object Contour

Here we apply the independent component analysis on the whitened contour $\tilde{Y}' = [x'(t') \ y'(t')]$. The steps involved in the algorithm are detailed below:

- Initialize a random matrix of weights W .
- Compute the intermediate matrix as:

$$W^+ = E\{\tilde{Y}'g(W^T \tilde{Y}')\} - E\{\tilde{Y}'g(W^T \tilde{Y}')\}W \quad (12)$$

where g is a non quadratic function and $E\{\}$ represents the maxima of the approximation of negentropy. For more details refer to [1].

- Let $W = W^+ / \|W^+\|$
- If not converged, then go back to b.

It is important to note that convergence means that the previous and current values of W have the same sign and the difference is below a certain permissible value.

By using the above procedure we have been able to find a matrix W' of weights that satisfies:

$$W' Y'(t') = W' A S(t') \approx S(t'), \quad A' = W'^{-1} \quad (13)$$

So we now use the inverse of the matrix W' to find S as per equation (3). The obtained source $S(t')$ will have the same statistical characteristics as the original source $S(t)$ but will only differ from it because of the random contour parameterization order.

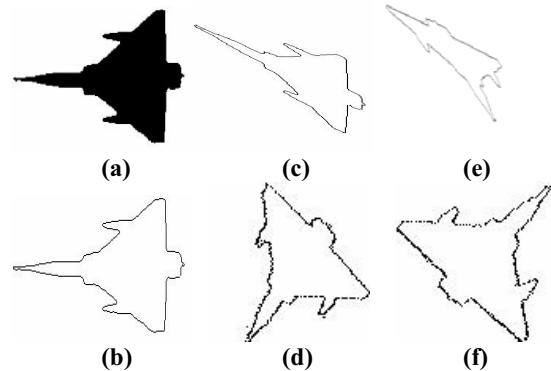


Figure 2 (a) Original Image (b) Parameterized boundary (c), (e) are affine transformed version of (a) and (d), (f) are the restored (normalized) counterparts obtained after applying above steps.

Figure 1 shows the complete system diagram and elaborates the above mentioned operations in a sequential and precise manner where as figure 2 and figure 3 demonstrate the output obtained after applying the above mentioned steps.

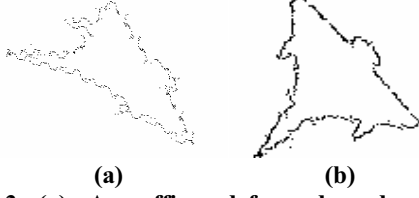


Figure 3 (a) An affine deformed and noise corrupted object contour (b) Noise reduced and affine normalized image obtained as a result of above operations.

Although using the above procedure we have been able to recover the contour of the object but the obtained independent components may have been inverted either along the parameterized x-axes or y-axes. As a result there are four possible cases $[x, y]$, $[x^t, y]$, $[x, y^t]$ and $[x^t, y^t]$ where x^t, y^t represent values in reverse order. However we can consider only one of the two cases $[x, y]$ and $[x^t, y^t]$ for invariant construction as the effect of inversion along both axes can be removed by using normalized cross correlation. So we are left with three cases and we construct invariants I_1 and I_2 proposed in the next subsection for each of the cases and use them while performing cross correlation.

3.3 Affine Invariant Functions

As a result of previous operations we have been able to remove translation, scaling and shearing distortions from the object contour besides reducing the effect of noise considerably which is introduced during the parameterization process because of incorrect segmentation. The only distortion we are left with is rotation. So in this third and final step we construct two invariants using the wavelet based conic equation for the restored object contour.

Conics have been used previously in computer vision to derive geometric invariant functions. For a point (x, y) from the restored object contour the conic can be expressed as the quadratic form [20]:

$$\begin{bmatrix} x & y \end{bmatrix} G \begin{bmatrix} x \\ y \end{bmatrix} = h, \quad \text{where } G = \begin{bmatrix} G_{11} & G_{12} \\ G_{12} & G_{22} \end{bmatrix} \quad (14)$$

where h is a constant and G is a symmetric matrix.

A wavelet based conic equation can be obtained from (14) using three dyadic levels $W_i x(t)$ and $W_i y(t)$ where W represents the wavelet transform and $i \in \{a, b, c\}$.

$$\begin{bmatrix} W_i x(t) & W_i y(t) \end{bmatrix} \zeta(t) \begin{bmatrix} W_i x(t) \\ W_i y(t) \end{bmatrix} = h$$

$$\text{where } \zeta = \begin{bmatrix} \eta_{11} & \eta_{12} \\ \eta_{12} & \eta_{22} \end{bmatrix} \quad (15)$$

An affine invariant function can then be defined as:

$$\eta_{a,b,c}(t) = \eta_{11}(t)\eta_{22}(t) - \eta_{12}^2(t) \quad (16)$$

The above function has been proven [5][6] to be equivalent to:

$$\eta_{a,b,c} = -f_{c,b}^4(t) - f_{a,c}^4(t) - f_{b,a}^4(t) + 2f_{a,c}^2(t)f_{c,b}^2(t) + 2f_{c,b}^2(t)f_{b,a}^2(t) + 2f_{b,a}^2(t)f_{a,c}^2(t) \quad (17)$$

where

$$f_{p,q} = A_p x(t)A_q y(t) - A_q x(t)A_p y(t) \quad (18)$$

The function in (17) is an invariant of weight four. We make use of the approximation coefficients of the wavelet transform while constructing the invariants I_1 and I_2 using (17) and the dyadic wavelet transform is implemented using the ‘‘A Trou algorithm’’ proposed by Mallat [19]. Figure 4 shows the plot of invariants I_1 and I_2 .

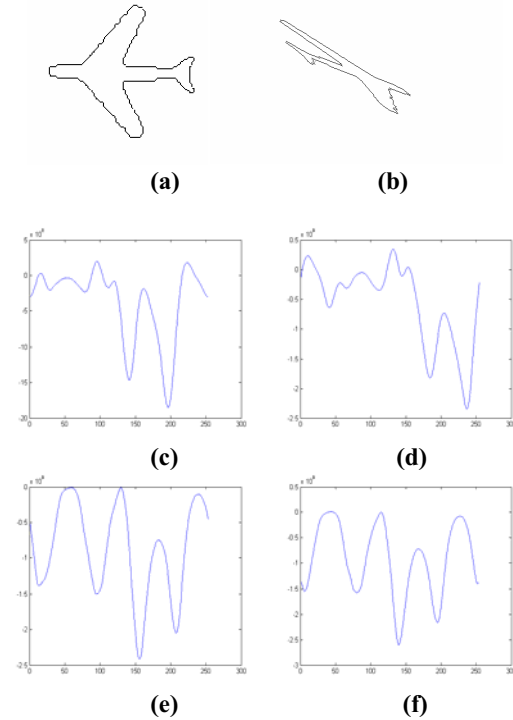


Figure 4 (a) Original Image. (b) Affine transformed image. (c), (d) shows invariant I_1 for images in (a) and (b). (e), (f) shows invariant I_2 for images in (a) and (b).

4 Experimental Results

The proposed technique was tested on a 2.4 GHz Pentium 4 machine with Windows XP and Matlab as the development tool. The datasets used in the experiments include the MPEG-7 Shape-B datasets, 10 aircraft images from [6] and English alphabets dataset. All the parameterized contours are resampled

to have the same length L of 256 data points. In the construction of the invariant I_1 and I_2 the approximations coefficients at level $\{3, 4, 5\}$ and $\{2, 4, 6\}$ are used, where as cubic spline filters are used for wavelet decomposition. Besides this we use normalized cross correlation for comparing two sequences A_k and B_k which is defined as:

$$R_{AB} = \frac{\sum_l \sum_k A_k B_{k-l}}{\sqrt{\sum_k A_k^2 \sum_k B_k^2}} \quad (19)$$

This section is divided into three parts first we demonstrate the stability of the two invariants against five different affine transformations then we provide a comparative analysis of the two invariants with the method in [3] and lastly we demonstrate the feature discrimination capability of the two invariants when compared to the method of Fourier descriptors.

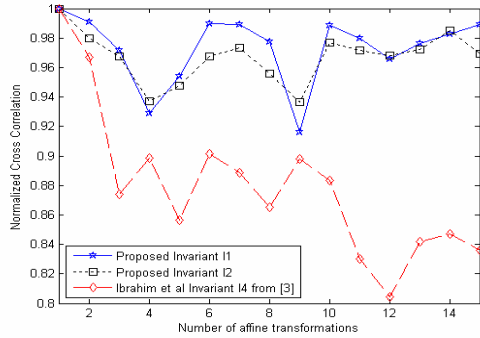


Figure 5 shows the comparison of invariant I_1 and I_2 with the method in [3]. The results are averaged over the MPEG-7 shape-B dataset.

Table 1 provides comparison of the invariants I_1 and I_2 in terms of the normalized cross correlation values against different affine transformations for the objects in figure 2(b) and figure 4(a) from the aircraft dataset. In the table following notation is used: Rotation (R) in degrees, Scaling (S), Shear (Sh) along x and y axis and Translation (T). The figures in brackets represent the parameters of the transformation.

Table 1 shows the normalized cross correlation values of the invariants after applying different affine transformations.

Transformation	Object 1 [2(b)]		Object 2 [4(a)]	
	I_1	I_2	I_1	I_2
Original Image	1.00	1.00	1.00	1.00
R(70), S(2,1)	0.9693	0.9571	0.9403	0.9335
R(135), S(2,3), T	0.9718	0.9709	0.9785	0.9596
R(45), Sh(2.05, 1.0), T	0.9369	0.9202	0.9035	0.9267
R(165), S(3,3), Sh(1,2), T	0.9845	0.9818	0.9148	0.9423
R(230), S(4,1), Sh(3,3), T	0.9376	0.9679	0.9217	0.9466

To further elaborate and demonstrate invariant stability figure 5 compares the proposed invariants I_1 and I_2 with [3] over a set of 15 affine transformations.

The results are averaged over the MPEG-7 shape-B dataset. Obtained results show a significant increase in performance as a function of increased correlation between the original and affine transformed images for the proposed invariants.

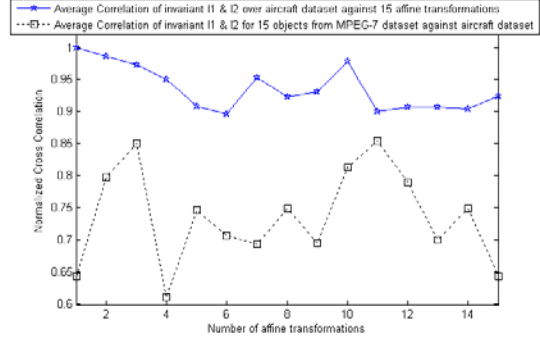


Figure 6 demonstrates the discrimination capability of invariant I_1 and I_2 using the aircraft and MPEG7 dataset.

Finally we demonstrate the feature discrimination capability of the proposed invariants using figure 6 and compare it with that of the Fourier Descriptors in figure 7. Figure 6 plots the result of correlation of the proposed invariants for the aircraft dataset and its fifteen affine transformed versions and correlation of fifteen objects and there affine transformed version from the MPEG-7 shape-B dataset with the aircraft dataset. The results have been averaged for I_1 and I_2 . For the invariants that can exhibit good disparity between shapes the two correlation plots should not overlap which has been the case for the proposed invariants I_1 and I_2 in figure 6. Figure 7 plots the above mentioned correlations using the method of Fourier Descriptors where the two correlation plots overlap significantly.

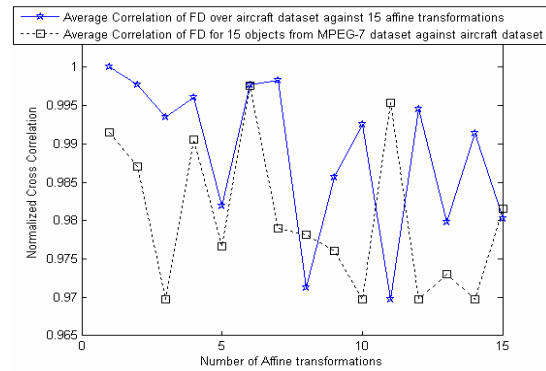


Figure 7 demonstrates the discrimination capability Fourier Descriptors using the aircraft and MPEG7 dataset.

It is important to mention here that a preprocessing step such as a smoothing operation applied on the object contour after restoration can significantly increase the correlation values, which at present has not been used to preserve the shape discrimination

power of the two invariants. Obtained results show significant reduction in error, thus validating the proposed approach.

5 Conclusion

In this paper we have presented a hybrid approach for invariant construction using the independent component analysis and wavelet based conic equation. Experimental results validate the use of an affine normalization technique as a preprocessor to the computation of invariant functionals. Beside this the use of dyadic wavelet transform after affine normalization added the much needed discriminative power to the proposed set of invariants. Presently, work is in progress to extend the framework to handle the projective group of transformations and estimation of the affine parameters, in future we intend to build an intelligent classifier for performing object recognition over a large dataset based on the proposed invariants.

6 Acknowledgment

The authors would like to thank National Engineering and Scientific Commission (NESCOM) for their financial support, GIK Institute of Engineering Sciences & Technology for facilitating this research and Temple University, USA for providing the MPEG-7 Shape-B dataset.

7 References

- [1] A.Hyvarinen, "Fast and robust fixed point algorithms for independent component analysis", *IEEE Transactions on Neural Network*, vol. 10 no.3, 1999.
- [2] A.Hyvarinen, E.Oja, "A fast fixed point algorithm for independent component analysis", *Neural Computations*, vol. 9, 1997.
- [3] I.E.Rube, M.Ahmed, M.Kamel, "Wavelet approximation based affine invariant shape representation functions", *IEEE Transactions on pattern analysis and machine intelligence*, vol.28, no.2, February 2006.
- [4] Q.Tieng, W.Boles, "An application of wavelet based affine invariant representation", *Pattern recognition Letters* vol. 16, 1995.
- [5] M.I.Khalil, M.Bayoumi, "Affine invariants for object recognition using the wavelet transform", *Pattern recognition letters*, no.23, 2002.
- [6] M.Khalil, M.Bayoumi, "A dyadic wavelet affine invariant function for 2D shape recognition", *IEEE Transactions on pattern analysis and machine intelligence*, vol.23, no.10 October 2001.
- [7] S.Manay, D.Cremers, B.Hong, A.Yezzi, S.Soatto, "Integral Invariants for shape Matching", *IEEE Transactions on pattern analysis and machine intelligence*, vol. 28, no. 10, October 2006.
- [8] H.W. Guggenheimer, "Differential Geometry", McGraw-Hill, New York, 1963.
- [9] K. Arbter, E. Synder, H.Burkhardt, G.Hirzinger, "Application of affine invariant Fourier descriptors to the recognition of 3D objects", *IEEE Transactions on pattern analysis and machine intelligence*, vol.12, no.7, 1990.
- [10] D.Zhao, J.Chen, "Affine curve moment invariants for shape recognition", *Pattern recognition*, vol.30, no.6 1997.
- [11] J.Flusser, T. Suk, "Pattern recognition by affine moment invariants", *Pattern recognition*, vol.26, no.1, 1993.
- [12] M.Petrou, A.Kadyrov, "Affine invariant features from the trace transform", *IEEE transactions on pattern analysis and machine intelligence*, vol.26, no.1, January 2004.
- [13] E.Rahtu, M.Salo, J.Heikkila, "Affine invariant pattern recognition using multiscale autoconvolution", *IEEE Transactions on pattern analysis and machine intelligence*, vol.27, no.6, June 2005.
- [14] Y.Lamdan, J.T.Schwartz, "Affine Invariant Model based object recognition", *IEEE Transactions on robotics and automation*, vol.6, no.5, October 1990.
- [15] T. Wakahara, K.Adaka, "Adaptive Normalization of handwritten characters using global-local affine transformations", *IEEE Transactions on pattern analysis and machine intelligence*, vol.20, no.12, December 1998.
- [16] I.E.Rube, M.Ahmed, M.Kamel, "Coarse to fine multiscale affine invariant shape matching and classification", *Proc of 17th International Conference on Pattern recognition*, 2004.
- [17] Z.Hauang, F.S.Cohen, "Affine invariant B-spline moments for curve matching", *IEEE Transactions on image processing*, vol.5, no.10, October 1996.
- [18] J.Mundy, A.Zisserman, "Geometric invariance in computer vision", MIT Press, Cambridge, MA.
- [19] S.Mallat, "A Wavelet Tour of Signal Processing" 2nd Edition, Academic Press, 1999.
- [20] I.Weiss, "Geometric invariants and object recognition", *International journal of computer vision*, vol. 10, no. 3, 1993.

VQ-Based Data Hiding in Images by Minimum Spanning Tree

Hung-Min Sun¹, King-Hang Wang², Hou-Wen Wang³, Chia-Yen Chen⁴

^{1,2,3}Department of Computer Science
National Tsing Hua University, Taiwan

⁴Department of Computer Science,
The Univeristy of Auckland, New Zealand

Email: ¹humsun@cs.nthu.edu.tw, {²khwang0, ³blark}@is.cs.nthu.edu.tw, ⁴yen@cs.auckland.ac.nz

Abstract

Image data hiding transmits secret information via side channel. The two conflicting properties: image quality and data capacity, are two major concerns in image data hiding, especially in compressed image format. In this paper, we seek a balance between the conflicts using vector quantization (VQ). Two approaches have been proposed based on coloring in minimum spanning tree. One of our schemes has shown to provide the best theoretic heuristic solution. Our results have significant advantages over other techniques as shown by the experimental results.

Keywords: image data hiding, vector quantization, minimum spanning tree

1 Introduction

Steganography is designed to carry secret information in side channel without anyone noticing. Image data hiding and watermarking [11] are two similar topics in steganography, but they have significant differences from the purposes they serve and the applications they are employed in. Watermarking is mainly used for protecting the metadata of digital contents. It provides the proof of the ownership of a digital content. The major concern of watermarking is its robustness. Image data hiding is employed to secretly carry information through a static image. A typical application of data hiding is in the military. It focuses on the capacity (the amount of data being hidden) and the imperceptibility (chances of being discovered that the image is loaded with information), which can also be explained as the image quality degraded by the embedding secret data. In contrast to watermarking, data hiding embeds lossless data in an image rather than lossy data. In addition, the data carried in data hiding are usually encrypted; that makes adversary unable to extract the hidden information; while some of the watermarks can be publicly retrieved.

Data hiding schemes are divided into spatial domain and frequency domain. A classic method in spatial domain is Least Significant Bit (LSB) substitution [1][7]. This method substitutes the LSB of every pixel in the image with the information it embeds. Methods working in spatial domain are generally faster and are able to carry larger amount of data. Methods in frequency domain apply discrete Fourier transform

(DFT) or discrete cosine transform (DCT) to transform the spatial signal to the frequency domain, where data is embedded in the coefficients of the DCT or DFT equations. Schemes designed in frequency domain are more robust against different types of image operations (like sharpening, contrast, adjustment, etc.).

Vector Quantization (VQ) [10] is a lossy compression technique in spatial domain. It divides the image into many small blocks and tries to represent each block with a master palette (codebook). This may significantly reduce the size of the original image.

Current trend is to handle images in a compressed format, yet, the distortion introduced by lossy compression has limited the room for data hiding. Therefore, it is a great challenge for researchers to develop an efficient data hiding scheme with VQ.

Some previous works have been proposed for data hiding with VQ [3][5][9]. These methods fail to achieve good image quality and high data rate simultaneously. Also, some of the schemes [4] further require the secret encoder and the secret extractor to share an additional codebook for embedding and extracting the secret. This assumption may not be feasible in many applications.

In this paper, we propose two data hiding schemes with VQ. The first approach is a simple scheme that is capable of hiding one bit per block and the second scheme is able to hide multiple bits per block. The contribution of our work is significant. Our first approach is the best theoretical heuristic solution in hiding a single bit per block, while the second

approach has a significant advantage over the work of [3]. These two approaches do not require the sender and the receiver to share an additional codebook. Furthermore, our algorithm allows VQ compression to be completely separated from secret embedding. This feature may facilitate lightweight devices in data embedding.

The organization of the paper is as follows. We review some related data hiding schemes in VQ in section 2. In section 3, we introduce the proposed schemes, followed by performance analysis in section 4. We conclude the paper in section 5.

2 Related Works

In this section, we take a quick look at VQ and review four previous approaches on data hiding. They are mean gray-level embedding method (MGLE), pair-wise nearest-neighbor embedding method (PNNE), principle component analysis method (PCA), and least significant bit substitution method (LSB).

2.1 Vector Quantization

VQ is a well known technique to compress images [10]. First, a codebook is generated and shared by the encoder and decoder. A codebook of size N contains 2^m vectors (codewords) $C = \{C_i | 1 \leq i \leq 2^m\}$. Each codeword is in j dimensions $C_i = \{C_{i,1}, C_{i,2}, \dots, C_{i,j}\}$. The LBG algorithm [8] is a classic method to train a representative codebook. To encode a gray-level image I , we first divide the image into many non-overlapped blocks $v_i = \{v_{i,1}, v_{i,2}, \dots, v_{i,j}\}$ where $I = \{v_1, v_2, \dots, v_k\}$. Each block contains n by n pixels ($j = n \times n$). Then, for every block, we find the closest codeword from the codebook and record their index. Therefore, each block of the image will be represented by an index. These indices will be sent to a receiver as a compressed image. The receiver can reconstruct the image by filling up codewords found in the codebook.

2.2 Mean Gray-Level Embedding method (MGLE)

MGLE appears in [3][4] as the intuitive approach in data hiding with VQ. All codewords in the codebook are sorted in advance according to their mean value. The mean value of each codeword c_i is calculated by $\bar{c}_i = (\sum_{s=1}^j c_{i,s}) / j$. Then, these codewords are divided into two sub-codebooks with the odd indices in '0'-codebook and even indices in '1'-codebook. As shown in Figure 1, we encode each block v_i with different sub-codebook, depends on the secret bit being embedded. If the i -th bit of the secret is '0', we will choose the '0'-codebook to encode the block v_i . The decode procedure is the same to the VQ decode. To extract the data embedded in the image, we simply search which sub-codebook the codewords belong to.

The codebook is sorted according to the mean value based on the belief that two codewords with similar mean value have a short Euclidean distance. Two close codewords should not be placed in a same codebook. Follow this logic, this sorting method tries to avoid any two close codewords being placed in the same codebook.

We shall see the reason to separate two close codewords in different sub-codebooks in section 3. However, the reader may realize that the difference between the mean values of two codewords has no implication on the distance between them. This can be justified by the example of the following two codewords: $\{10, 0\}$ and $\{0, 10\}$ where both of them have the mean value 5 but the distance in between is $\sqrt{(10-0)^2 + (0-10)^2} = 10\sqrt{2}$. The mean value of $\{4, 0\}$ and $\{0, 0\}$ is different by 2 while their distance is only 2. These two examples prove that the difference of mean values of two codewords does not have absolute implications on the distance between them.

2.3 Pair-wise Nearest-Neighbor Embedding method (PNNE)

PNNE [4] is an improvement of MGLE. The design philosophy of PNNE is to pair up nearest codewords together. In PNNE, we repeatedly select two codewords which are closest to each other, label one of them as '0' and the other one as '1'. The encoding and decoding schemes of PNNE are exactly the same as MGLE. They are only different in the division of sub-codebooks.

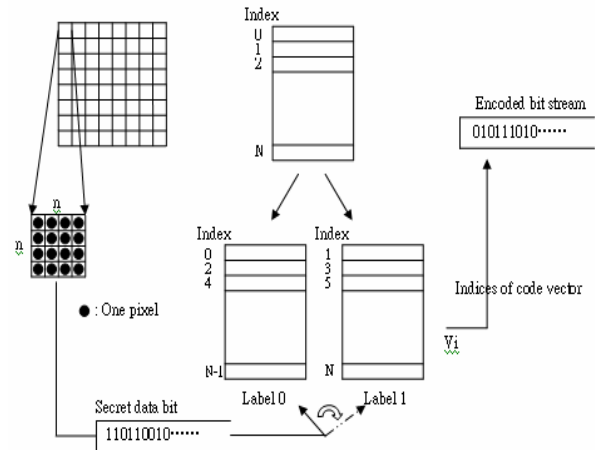


Figure 1: The encoding procedure used in MGLE, PNNE, and PCA.

2.4 Principle Component Analysis (PCA)

Principle component analysis (PCA) [6] is a famous algorithm widely used in pattern recognition and data analysis. PCA clusters the input vectors by projecting them into a K -dimension space.

In [2], the authors proposed that the codebook should be sorted using PCA instead of the mean value of

codewords. The motivation of PCA method is somehow the same as PNNE – to cluster near codewords together. This method can be generalized in hiding multiple bits in a single block [2].

2.5 Least Significant Bit (LSB)

LSB method [7] first appeared in uncompressed domain. It substitutes the least significant bit of each pixel with the secret information. Later, it has also been introduced in VQ domain [2]. By sorting the codebook with some mechanisms, after encoding the image using standard VQ technique, we substitute the least significant bit of each code vector. In some cases, we may further stuff more bits into each codeword by substituting the last two or three bits of the code vector. As a consequence, this will further degrade the image quality.

The method of sorting codebook, like MGLE and PCA, can be used with LSB method. Later in this paper, we refer MGLE_LSB and PCA_LSB as the method that combines MGLE and PCA with LSB method respectively.

3 The Proposed Approach

3.1 Motivation

Consider a toy example shown in the Figure 2. We have four codewords $C = \{C_1, C_2, C_3, C_4\}$ in a codebook. To simplify the problem, we further assume that these codewords lie on a straight line with equal distance apart. More specifically, the distance D_{ij} between C_i and C_j is defined as $D_{ij} = K \cdot |i - j| \exists$ constant K and $1 \leq i, j \leq 4$.

Before embedding data into an image, we have to separate these codewords into two sub-codebooks G_0 and G_1 . There are $16 - 2 = 14$ ways (excluding two with empty set) to do so. We compare two ways to separate these codewords. The first one sets $G_0 = \{C_1, C_2\}; G_1 = \{C_3, C_4\}$ and the second one sets $G'_0 = \{C_1, C_3\}; G'_1 = \{C_2, C_4\}$. Let P_i be the distribution of the distance between a codeword C_i and a random code vector sampled from original image. We assume that all P_i 's are identical; that implies any random vector has equal probability to be closest to any codeword. We claim that the second approach to separate the codewords has a better performance than the first approach, which is proven in the following lemma.

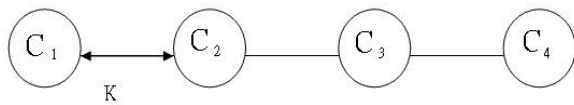


Figure 2: Example to illustrate the importance of codeword classification in data hiding

Lemma 1: The average distortion in encoding with $\{G'_0, G'_1\}$ is less than encoding with $\{G_0, G_1\}$.

We learn from the toy example and Lemma 1 that the penalty induced in data hiding is controlled by “how far” away the alternating codeword is. If the alternating codeword of C_i is C_3 , which means we are now using $\{G_0, G_1\}$, the longer distance between C_1 and C_3 would cost us more distortion penalty. Therefore minimizing the distance between each pair of “alternating” codewords is very important.

To formalize our goal in designing a good data hiding scheme, we can generalize the distortion penalty equation from the toy example as listed as below:

$$T = \left(\frac{1}{|G_2|}\right) \sum_{C_i \in G_2} \sum_{U=C_i} \Pr(U=C_i) \left(\min_{C_j \in G_1} D_{ij}\right) \\ + \left(\frac{1}{|G_1|}\right) \sum_{C_i \in G_1} \sum_{U=C_i} \Pr(U=C_i) \left(\min_{C_j \in G_2} D_{ij}\right)$$

Minimization of the above term is equivalent to the minimization of the average distance between each pair of alternating codewords. This can be optimized by using minimum spanning tree which guarantees that every codeword has an alternating codeword in the other sub-codebook. Owing to page limitation, the reduction is omitted.

3.2 Our Proposed Scheme

We first describe the simple scheme that embeds only a single bit in each block. Roughly speaking, our approach is to build a minimum spanning tree with the codewords being vertices and their distances being edges. Then, we color the spanning tree with the requirement that no adjacent vertices have the same color. These colors represent the information bit carried by the codewords. To encode a block, we simply choose its nearest codeword or the corresponding alternative, depends on the information to be hidden, to represent the block.

Detailed procedures of the algorithm are illustrated below:

Generation of codebook: (output: a colored spanning tree T)

- 1) Generate a codebook $C = \{C_i \mid 1 \leq i \leq 2^w\}$ with 2^w of codewords and each codeword is $n \times n$ large.
- 2) Build a minimum spanning tree based on the Euclidean distance amongst codewords
- 3) Paint C_1 with color ‘0’ and add C_1 into a queue Q
- 4) Do the following until all Q is empty.
 - I. Pop the first element C_k from Q
 - II. Set $p' = (p + 1) \bmod 2$ where p is the color of C_k
 - III. Color all the uncolored neighbors of C_k as p' and add them into Q

Embedding: (Input: a stream of block $B = \{b_1, b_2, b_3, \dots \mid b_i \in R^{n \times n}\}$, a codebook

$C = \{C_i \mid 1 \leq i \leq 2^w\}$, a colored spanning tree T , a stream of secret bits $S = \{s_1, s_2, s_3, \dots \mid s_i \in \{0,1\}\}$. Output: a stream of indices $D = \{d_1, d_2, \dots \mid \forall d_i \in C\}$)

- 1) Find the nearest codeword of b_i . Name it as C_k
- 2) If the color of $C_k = s_i$, encode b_i as C_k ; Otherwise, find the nearest neighbor of C_k from the spanning tree and encode b_i as that neighbor.

Decode and extract: (Input: a stream of index $D = \{d_1, d_2, \dots \mid \forall d_i \in C\}$, codebook C .

Output: blocks $B' = \{b'_1, b'_2, b'_3, \dots \mid b'_i \in R^{n \times n}\}$, a stream of secret bits $S' = \{s'_1, s'_2, s'_3, \dots \mid s'_i \in \{0,1\}\}$)

- 1) Build the colored minimum spanning tree using the method mentioned above
- 2) Rebuild a block b_i by replacing each index d_i by the codeword found on the codebook.
- 3) Extract the secret bit s'_i by looking up what color d_i is from the colored spanning tree.

Since the Euclidean distance of codeword follows triangular inequality, there should have a unique minimum spanning tree that guarantees each codeword directly connects to its nearest neighbor. Also, there is no cycle in a minimum spanning tree; therefore, our algorithm would not paint two adjacent nodes with the same color. As a result, step 3) of *Embedding* will encode the secret bit properly, even if the color of C_k does not equal to s_i .

The quality of an embedding image is solely reflected by its PSNR value, which is in fact the Euclidean distance of the recovered image and original image. Despite the random distribution of source image that brings uncertainties to different embedding algorithms, this embedding scheme promises that the alternative of every codeword has a minimum distance apart from it; that also suggests the best theoretic heuristic solution for hiding a single bit in each block.

We describe the extension of our algorithm that hides multiple bits in a block. Assuming that we are going to hide r -bits secrets in a block, the *Decode and Extract* procedures are same as the signal-bit hiding scheme except the color of the spanning is no longer binary. We explain the algorithms for *Generation of Codebook* and *Embedding* in details here:

Generation of Codebook: (Output: a 2^r -colored spanning tree T)

- 1) Generate a codebook $C = \{C_i \mid 1 \leq i \leq 2^w\}$ and build a minimum spanning tree based as describe in the basic scheme.
- 2) Paint C_1 with color '0' and add C_1 into a queue Q
- 3) Create two tables called $Global(i)$ and $Local(i,j)$ where $0 \leq i \leq 2^r - 1, 1 \leq j \leq 2^w$. Initially all entries are zero.
- 4) Do the following until all Q is empty.

- I. Pop the first element C_k from Q and set p is the color of C_k
- II. For each C_j being the uncolored neighbors of C_k
 1. Name the set of i as I that minimizes $Local(i,k)$ where $i \neq p$
 2. If $|I| > 1$, find i that minimizes $Global(i)$
 3. If there are more than one i that minimize both $Local(i,k)$ and $Global(i)$, then select the least i .
 4. Set $Global(i) = Global(i) + 1$ and $Local(i,k) = Local(i,k) + 1$
 5. Color C_j as i and add C_j into Q

Embedding: (Input: a stream of block $B = \{b_1, b_2, b_3, \dots \mid b_i \in R^{n \times n}\}$, a codebook $C = \{C_i \mid 1 \leq i \leq 2^w\}$, a colored spanning tree T , a stream of secret bits $S = \{s_1, s_2, \dots \mid s_i \in \{0, \dots, 2^r - 1\}\}$. Output: a stream of indices $D = \{d_1, d_2, \dots \mid \forall d_i \in C\}$)

- 1) Find the nearest codeword of b_i . Name it as C_k
- 2) If the color of $C_k = s_i$, encode b_i as C_k ; Otherwise, find the nearest codeword from C_k with the same color as s_i from the spanning tree and encode b_i as that codeword.

This algorithm is an extension of our simple scheme. The underlying theory is the same – to minimize the distance between codewords and their alternating codewords. As we can see, if the number of colors increases, the average distance between codewords and their alternating codewords will also increase. This degrades the image quality of the embedded image, but in exchange, allows more data to be carried by the image.

We note that the two algorithms can be further improved by fully searching the best codewords from the candidates set, rather than just selecting the alternative codewords to replace the nearest codeword. However, from our experience, these improvements are not significant.

As we have mentioned above, the data embedding scheme can be totally separated from the VQ compression. On the inputs of the codebook, and the VQ-compressed image, one may embed secret data using the algorithm stated above, by simply replacing the index by the desired alternative codewords. Pre-computation of colored minimum spanning tree and alternative codewords mapping can be done at a backend server in advance, to allow a lightweight device embed secret in an image efficiently.

4 Performance

The performance of distortion is measured by PSNR. PSNR is defined as:

Table 1: Hiding 16384 bits ($r=1$) in two testing images

PSNR	VQ (no hiding)	Ours	PCA	PNNE	MGLE
Tiffany	30.96	28.68	27.38	27.53	26.68
F-16	27.59	26.17	25.70	25.91	25.36

Table 2: Hiding bits = $r \times 16384$ (bits) in two testing images

PSNR	Ours		PCA_LSB		MGLE_LSB	
	Tiffany	F-16	Tiffany	F-16	Tiffany	F-16
$r = 1$	28.68	26.17	24.40	22.34	24.43	22.12
$r = 2$	24.95	23.91	22.66	20.85	22.73	20.80
$r = 3$	22.17	21.66	21.62	20.00	21.63	19.96

$$PSNR = 10 \times \log_{10} \frac{(2^b - 1)^2}{MSE} \text{ dB}$$

$$MSE = \frac{\sum_{i=1}^K (o_i - r_i)^2}{K}$$

Here, K is the size of image, 2^b is the size of the codebook, o_i is the i -th pixel value of the original image, and r_i is the i -th pixel value of the compressed image. We generate the codebook using LBG algorithm with Tiffany (Figure 3(a)) in the training set and F16 (Figure 3(b)) not in the training set. The tested images are of 512×512 , 8-bits gray-level. The codebook size is set to 256, each block size is 16 (4×4). Experimental results, in terms of PSNR, are shown in Table 1 and Table 2. In Table 1, we hide a single bit ($r = 1$) in each block of totally 16K bits of randomly generated secret. We compare our result with MGLE, PCA, and PNNE with Tiffany and F-16. For the same embedding capacity, we have achieved the best image quality. Next, we perform another test with our extended scheme with various numbers of bits hidden in each block. We compare this to the PCA_LSB and the MGLE_LSB in Table 2. Again, we have significant performance over PCA_LSB and MGLE_LSB. Figures 4 and 5 show some pictorial results of different schemes.

5 Conclusions

In this paper, we propose two image data hiding methods with VQ using minimum spanning tree. As shown in the experimental results, we have achieved best image quality among all of the existing schemes. By using coloring minimum spanning tree, we reduce the penalty when embedding secret in images. In the future, we will try to adapt this technique to hide data in images which are transformed in frequency domain. We will also try to cope this with industrial standards, such as JPEG.

Acknowledgements

The authors wish to acknowledge the anonymous reviewers for valuable comments. This research was supported in part by the National Science Council,

Taiwan, under contract NSC 95-2221-E-007-021.

References

- [1] C. K. Chan and L. M. Cheng, "Hiding data in images by simple LSB substitution," *Pattern Recognition*, vol. 37, pp. 469-474, March 2004
- [2] C. C. Chang, D. C. Lin, and T. S. Chen, "An improved VQ codebook search algorithm using principal component analysis," *Journal of Visual Communication and Image Represent*, vol. 8, pp. 27-37, March 1997
- [3] C. C. Chang, and P. Y. Lin, "A compression-based data hiding scheme using vector quantization and principle component analysis," *3rd International Conference on Cyberworlds (CW 2004)*, pp. 369-375, 2004
- [4] W. C. Du, and W. J. Hsu, "Adaptive data hiding based on VQ compressed images," *IEE Proc., Vis. Image Signal process*, vol. 150, No. 4, pp. 233-238, August 2003
- [5] Y. C. Hu, "Gray-level image hiding scheme based on vector quantization," *IEE Electronic Letter*, vol. 39, No. 2, pp. 203-203, January 2003
- [6] I.T. Jolliffe, "Principle Component Analysis," New York: Springer-Verlag, 1986.
- [7] W. N. Lie, L. C. Chang, "Data hiding in images with adaptive numbers of least significant bits based on the human visual system," in *Proc. IEEE Int. Conf. on Image Processing*, vol. 1, pp. 286-290, 1999.
- [8] Y. Lined, A. Bozo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, vol. 28, pp. 84-95, Jan 1980
- [9] Z. M. Lu, W. Xing, D. G. Xu, and S. H. Sun, "Digital image watermarking method based on vector quantization with labeled codeword," *IEICE transactions on information and systems*, vol. E86-D, pp. 2786-2789, Dec 2003

- [10] R. M. Grey, "Vector quantization", *IEEE ASSP Magazine*, pp. 4-29, April 1984
- [11] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding – A survey," in *Proceedings of the IEEE, special issue on*

protection of multimedia content, vol. 87, pp. 1062-1078, July 1999



(a) Tiffany



(b) F-16

Figure 3 Images for Experiment



(a) VQ (no hiding), PSNR=30.96 (b) Ours, PSNR=28.68 (c) PNNE, PSNR=27.53 (d) MGLE, PSNR=26.68

Figure 4 Experimental results for embedding single bit per block



(a) VQ, PSNR=27.59 (b) PCA_LSB, r=1 (c) PCA_LSB, r=2 (d) PCA_LSB, r=3



(e) Ours, r = 1

(f) Ours, r = 2

(g) Ours, r = 3

Figure 5 Experimental results for embedding more than one bit per block

Morphology-based Stable Salient Regions Detector

E. Rangelova and E.J. Pauwels

Center for Mathematics and Computer Science (CWI), Amsterdam 1098 SJ, The Netherlands.

Email: ely@cwi.nl

Abstract

In this paper we describe a new detector of salient regions in an image. It is a generalization of the morphology-based binary pattern detector introduced in [1]. The detected regions are stable and salient over a range of image gray-tone cross-sections. These morphology-based stable salient regions (MSSR) are perceived as discriminating by humans and under certain conditions are affine-covariant. The MSSR detector is compared to two detectors, currently performing best on structured scenes under different transformations in the framework provided by Mikolajczyk et al [2]. Results show that the MSSR is similar with respect to repeatability and matching scores and significantly outperforms the others on perceived saliency of the detected regions, suited for object recognition and photo-identification tasks.

Keywords: saliency detection, affine covariant regions, morphology, photo-identification

1 Introduction

The task of finding reliable correspondences between two images of the same scene taken from different viewpoints and possibly under different acquisition conditions is a difficult, but important step in many applications such as image retrieval from large databases [3], wide baseline matching for stereo pairs [4, 5], object recognition [6], etc. In a typical matching application the first step is detecting a number of “salient” regions in each image independently. In this context we will use the term *salient* to indicate an image region that is reliably identifiable and stable with respect to a wide range of image transformations.

The detected regions should change *covariantly* (they are often called *invariant* [5, 7, 8]) with the transformation relating the two images [2]. For viewpoint changes *affinity* is often a sufficient model. A comparison of six state-of-the-art affine covariant region detectors is presented in [2]. The *Harris-affine* and *Hessian-affine* detectors are based on affine normalisation around Harris and Hessian points [7]. The ‘*maximally stable extremal regions*’ (*MSER*) method detects areas brighter or darker than the background via a threshold selection process. The *intensity-based* (*IBR*) detector starts with intensity extrema and explores the image along emanating rays, finding the extrema in the intensity changes [5]. The *edge-based* (*EBR*) detector moves from Harris corner points along nearby Canny edges identifying affine-covariant parallelograms [5]. The *salient regions* detector finds the maxima of the entropy of intensity values PDF over the scale and ellipse

spaces [8]. The conclusion in [2] is that “there does not exist one detector which outperforms the other detectors for all scene types and all types of transformations”. For *structured* scenes, containing homogeneous regions with distinctive boundaries, the *MSER* and *IBR* detectors perform best as they analyse the image isocontours directly (similarly to a watershed algorithm [9]).

We contend that simplifying the problem from a gray-tone/colour to binary saliency analysis provides a flexible framework. Similarly to *MSER*, we propose to analyse the image isocontours by decomposing the image into binary cross-sections and computing four saliency maps for each. They are combined into a final map such that the retained regions would be recognised by humans as salient in object recognition context. An example application is individual *photo-identification* of animals based on their natural markings [1].

There are two main contributions in this paper: an extension of the morphology-based binary saliency detector which we have introduced in [1], as well as a rule for reducing the number of output regions. Second, we also extend the detector to gray-tone/colour images of structured scenes. Not all detected regions are affine-covariant, but all are stable and salient.

In Section 2 we describe the binary saliency detection followed by its extension to gray-tone/colour images in Section 3. The evaluation of the performance of the Morphology-based Stable Salient Region (MSSR) detector is compared to *MSER* and *IBR* in Section 4 and the conclusions are given in Section 5.

2 Binary Salient Regions Detection

We contend that the saliency perceived in binary images $\mathbf{B} : \mathcal{D} \subset \mathcal{Z}^2 \rightarrow \{0, 1\}$ (1-white, 0-black) is only due to the spatial layout of the regions within the image. For instance, for the binary image in figure 1 the regions perceived as salient are the isolated holes and islands as well as the protrusions and indentations. If we designate the 1-regions as foreground (FG), then a *hole* (aka. *Inner Salient Structure (ISS)*) is a background (BG) region that cannot be connected to the image border. A *protrusion* (a.k.a. *Boundary Salient Structure (BSS)*) is defined to be a set of FG pixels such that if it's pinched off from a significant white connected component \mathcal{B}^1 , its boundary will increase by less than $2\pi r$ pixels (where r is the radius of the structuring element (SE), see below). A significant connected component (CC) is a CC with area (number of pixels) proportional to the area of the image by a factor of Λ : $\text{card}(\mathcal{B}^1) \geq \text{card}(\mathbf{B})/\Lambda$. Reversing the role of BG and FG yields the *islands* and *indentations* respectively. These definitions are presented formally in table 1.

Mathematical morphology tools are well suited for dealing with salient structures as described above [9]. In particular, a *hole filling* operation $\bullet(\cdot)$ can be used to pick up the holes if applied on the set of all white pixels \mathbf{B}_1 intersected with the set of all black pixels \mathbf{B}_0 :

$$S_{01}^i = \rho_{01}^i(\mathbf{B}) = \bullet(\mathbf{B}^1) \cap \mathbf{B}^0. \quad (1)$$

Islands can be obtained either similarly by reversing FG and BG or by identifying all non-significant white CCs:

$$S_{10}^i = \rho_{10}^i(\mathbf{B}) = \bullet(\mathbf{B}^0) \cap \mathbf{B}^1 = \mathbf{B}^1 \setminus \bigcup_j \mathcal{B}_j^1. \quad (2)$$

The morphological *opening* operator (based on SE E - disk with radius r) $\gamma_E(\mathbf{B})$ generally smooths an object's contour eliminating thin protrusions. As a consequence, protrusions can be highlighted by subtracting the opened FG CC from the original (this is known as the *white tophat transform* $WTH_E(\mathbf{B}) = \mathbf{B} - \gamma_E(\mathbf{B})$):

$$S_{10}^b = \rho_{10}^b(\mathbf{B}) = \bigcup_j WTH_E(\mathcal{B}_j^1). \quad (3)$$

Morphological *closing* $\phi_E(\mathbf{B})$ tends to narrow breaks and long thin indentations. Therefore, indentations can be picked up by applying the *black top hat* (BTH) transform $BTH_E(\mathbf{B}) = \phi_E(\mathbf{B}) - \mathbf{B}$:

$$S_{01}^b = \rho_{01}^b(\mathbf{B}) = \bigcup_j BTH_E(\mathcal{B}_j^1) \quad (4)$$

or by applying the WTH to the reversed image.

The binary salient operator ρ is defined via:

$$\rho = \gamma_\lambda \circ (\rho_{01}^i \cup \rho_{10}^i \cup \rho_{01}^b \cup \rho_{10}^b), \quad (5)$$

where the *area opening* operator γ_λ removes isolated regions smaller than λ pixels.

The hole-filling operation does not depend on the size of the SE, therefore the ISS can be detected as affine covariant regions. This is not the case for the scale-invariance of the detected BSS. We propose choosing a small r which also minimises the inaccuracy of maximum $2\pi r$ along the "disconnection boundary" $\partial(\mathcal{B}^f \setminus S_{fb}^b)$. All detected salient regions \mathbf{S} have accurate boundaries as there is no smoothing involved.

Figure 1 illustrates the saliency operator on a synthetic binary image (left) with dimensions 100×100 pixels. The parameters are: $\Lambda = 100$, $r = 5$, $\lambda = 10$. The detected regions are shown overlaid on the original image and are colour coded: S_{01}^i (holes) -in blue, S_{10}^i (islands)- in yellow, S_{01}^b (indentations) -in green and S_{10}^b (protrusions)- in red.

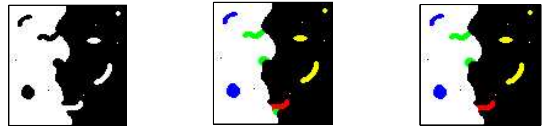


Figure 1: Binary salient regions detection.

If adjacent, the two types of BSS are in a sense complementary to each other, therefore redundant. For example in figure 1 (middle) a region encoded in red competes with a region encoded in green. A human observer would prefer to keep only the red region. We propose the winning region to be the one with larger eccentricity and the other will be removed (figure 1, right). In a natural image the number of detected salient regions would be reduced significantly by choosing the winning subsets from the two types competing BSS.

3 Gray-tone Salient Regions Detection

3.1 Gray-tone Saliency

Any gray-tone image $\mathbf{I} : \mathcal{D} \subset \mathcal{Z}^2 \rightarrow \{0, 1, \dots, t_{max}\}$, ($t_{max} = 2^n - 1$ is the maximum gray value encoded by n bits; we consider $n = 8$, $t_{max} = 256$) can be decomposed into cross-sections at every possible gray tone level t : $\mathbf{I} = \sum_{t=1}^{t_{max}} CS_t(\mathbf{I})$, ($CS_0(\mathbf{I}) = \mathbf{I}$). Obtaining a cross-section at level t is equivalent to thresholding the image and setting all pixels with values below the threshold t to 0 and the ones above- to 1, hence $CS_t(\mathbf{I})$ is a binary image.

ISS	A CC $S_{fb}^i = \{\mathbf{p} \in \mathcal{D}, \forall \mathbf{p} = f, \forall \mathbf{q} \in \partial S_{fb}^i, \mathbf{q} = b, \mathbf{q} \notin \partial \mathbf{B}\}$,
2 types of ISS	$S_{10}^i, f = 1, b = 0$ -white FG pixels on black BG (<i>islands</i>). $S_{01}^i, f = 0, b = 1$ -black FG pixels on white BG (<i>holes</i>).
BSS	$S_{fb}^b : \{\mathbf{p} \in S_{fb}^b \subset \mathcal{B}^f, \forall \mathbf{p} = f, \mathbf{q} \in \partial S_{fb}^b \subset \partial \mathcal{B}^f, \forall \mathbf{q} = b\}$, $card(\partial \mathcal{B}^f) - card(\partial(\mathcal{B}^f \setminus S_{fb}^b)) < 2\pi r$
2 types of BSS	$S_{10}^b, f = 1, b = 0$ - subset of significant white CC with black boundary (<i>protrusion</i> from white CC/ <i>indentation</i> in black CC). $S_{01}^b, f = 0, b = 1$ - subset of significant black CC with white boundary (<i>indentation</i> in white CC/ <i>protrusion</i> from black CC).
Salient regions	$\mathbf{S} = \mathbf{S}^i \cup \mathbf{S}^b$ - union of all ISS $\mathbf{S}^i = S_{01}^i \cup S_{10}^i$ and all BSS $\mathbf{S}^b = S_{01}^b \cup S_{10}^b$.

Table 1: Binary saliency definitions used in Section 2.

We define a morphology-based stable salient region (MSSR) in a gray-tone image as a region whose underlying binary salient region has a stable support over range of cross-sections. The saliency detection is then simplified to the binary case, where the perception of saliency is easier to model. Since the underlying binary salient regions are of 4 possible types, the same holds for the MSSR. These types will be detected separately and combined in a final saliency map.

In [4] MSERs are defined as the set of all connected components which are stable (do not change their area significantly) over several cross-sections. We believe that while the definition of MSSR is similar to MSER (or IBR), it is more general in two aspects. Firstly, MSSR include salient regions of type BSS (at the expense of partial affine covariance) which are undetectable for MSER or IBR, and secondly, the detection of MSSR can be easily extended to images where the saliency is not due to intensity, but to texture, colour, orientation, etc.

Our approach also fits well with Itti’s saliency-based model of visual attention [10], which is based on the of the visual system of primates. According to the model, the visual input is decomposed into different feature maps which are combined in a bottom-up manner into a master saliency map.

3.2 MSSR Detector

The actual detection of the salient regions in an input gray-tone image \mathbf{I} proceeds through a number of steps. First we create four empty maps $\mathbf{M}_{01}^i, \mathbf{M}_{10}^i, \mathbf{M}_{01}^b$ and \mathbf{M}_{10}^b of the same size as the input image, one for each type of saliency (i.e. holes, islands, indentations and protrusions) Then we use the gray-tone input image \mathbf{I} to produce an ordered stack of binary cross-sections for $t = 1, \dots, t_{max}$. For each cross-section we detect the four types of binary salient regions (Section 2) and accumulate the evidence in the corresponding map. For the ISS $\mathbf{M}_{fb}^i = \sum_t \rho_{fb}^i(CS_t(\mathbf{I}))$, using equation (1) for holes ($f = 0, b = 1$) and (2) for islands ($f = 1, b = 0$). Analogous, for the BSS

$\mathbf{M}_{fb}^b = \sum_t \rho_{fb}^b(CS_t(\mathbf{I}))$, using equation (4) for indentations and (3) for protrusions.

We then use a data-driven threshold to suppress weak responses, thus converting the four maps into binary maps. Finally, we post-process the BSS-maps: if a protrusion is adjacent to an indentation then only the one with the largest eccentricity (most salient) is retained in the updated BSS-maps ($\widehat{\mathbf{M}}_{10}^b$ and $\widehat{\mathbf{M}}_{01}^b$). After the reduction of regions (whenever necessary), the final saliency map \mathbf{M} is obtained by taking the maximum response of all maps $\mathbf{M}_{01}^i, \mathbf{M}_{10}^i, \widehat{\mathbf{M}}_{01}^b$ and $\widehat{\mathbf{M}}_{10}^b$.

4 Performance Evaluation

The performance of a region detector is measured by the repeatability criterion i.e. how well does the detector determine corresponding scene regions. The overlap error between two regions is defined by the ratio intersection/union $E = 1 - \frac{intersection}{union}$. Two regions are considered matching if $E < 40\%$ [2]. The *repeatability score* is the ratio between the number of region correspondences and the smaller number of regions detected in the common part of a pair of images (N_{min}). A measure in practical application is the *matching score*, i.e. the ratio between the number of correct matches and N_{min} .

We compare the performance of the MSSR detector to the best performing for structural scenes IBR and MSER detectors [2].

4.1 Affine Covariance

The robustness to different affine geometric and photometric transformations can be measured if the mapping between an image and a reference image is known. The output shape of most detectors is an ellipse. However, for MSER and MSSR, it is not and equivalent ellipses are constructed in affine-covariant manner (having the same first and second order moments as the original regions). Example of original MSSR output is given in figure 7, in all other figures, the ellipse representation is

used. To compute the overlap between two elliptical regions, they are first normalised to compensate for different size and then one region is projected into the reference image by using the known affine transformation.

Figure 2 shows the results on two synthetic binary images simulating all 4 types of saliency. The second image is affine-transformed version of the first. The equivalent ellipses (scaled with a factor of 2) of the detected regions by MSER and MSSR are shown in red. For each detected region also its normalised version is shown. The MSSR detector finds all perceptually salient regions, while MSER misses the BSS regions as predicted (Section 3). Both MSER and our detector demonstrate affine covariance for the ISS regions. The two types of BSS regions detected by MSSR also have good visual correspondence.

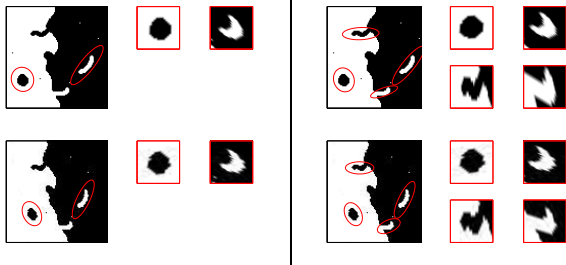


Figure 2: Affine covariance of the region detectors. Left column: MSER, right column: MSSR.

We have used all the data sets and evaluation software presented in [2] to compute the repeatability and matching scores under different transformations (available at <http://www.robots.ox.ac.uk/~vgg/research/affine/>). Two of the test sequences are shown in figure 3 (first and last images). The graffiti sequence (viewpoint change) has 6 images with view angle changing by 10° from 20° to 60° . The illumination changes in the Leuven sequence are decreasing light (by varying the camera aperture).

Figure 4 illustrates the affine-covariance of the detected regions of the 3 detectors with a detail from two images from the graffiti sequence. The MSSR detector typically gives the smallest number of detected regions which can be of great advantage at the matching step. The MSSR regions are also perceived as salient due to the “semantic” knowledge in the definition of the underlying binary saliency.

The repeatability and matching scores were computed for all test data sequences. As illustration, the results for the graffiti sequence are summarised in figure 5 and for the Leuven sequence in figure 6. According to these criteria the MSER detector performs best. For the viewpoint change it is followed



Figure 3: Data set- structured scenes. First row: graffiti sequence-viewpoint change, second row: Leuven sequence: illumination change. The left most image is used as the reference image.

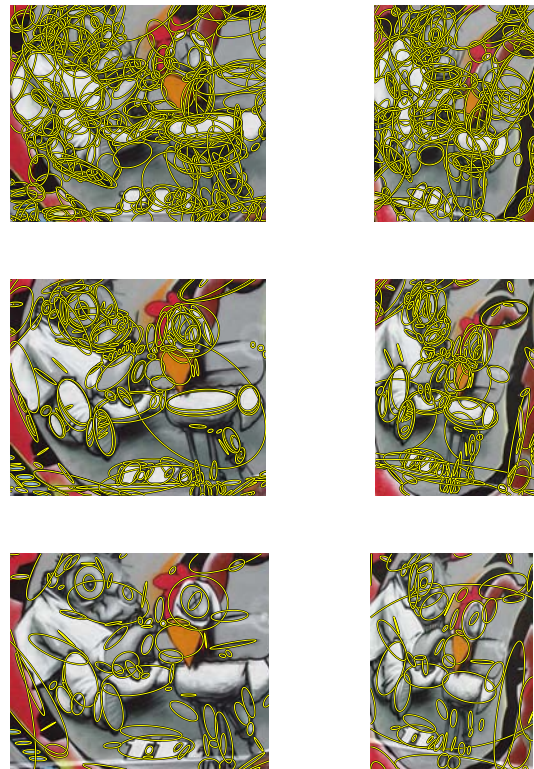


Figure 4: Viewpoint change. Detail from the reference image (left) and 30° viewpoint change (right) of the graffiti sequence (fig. 3, row 1.). From top to bottom row: IBR, MSER, MSSR.

by the MSSR, while for the illumination change, the MSSR is comparable to IBR in respect to repeatability with better performance in the more relevant for object retrieval matching score. Note that for recognising objects, repeatability might not be the best measure for suitability for classification. The matching score is more relevant in this context, but is dependant on the region descriptor.

In these experiments Lowe’s SIFT descriptor [11] on the gray values within each normalised region was used.

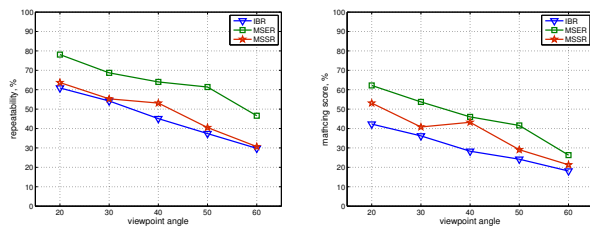


Figure 5: Viewpoint change: graffiti sequence (fig. 3, row 1.). Repeatability and matching scores.

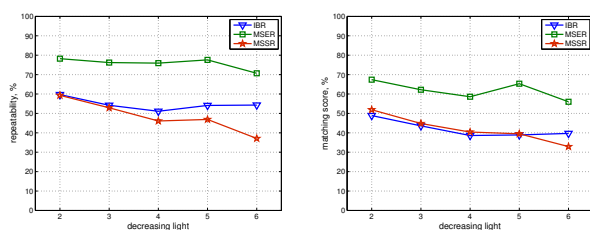


Figure 6: Illumination change: Leuven sequence (fig. 3, row 2.). Repeatability and matching scores.

4.2 Perceptual Saliency

In object recognition application, it is very important that regions perceived as salient by humans are also picked up by a detector. An interesting application is the individual photo-identification of marine mammals which utilises the unique natural markings captured by photographing the dorsal fins or flukes (i.e. tails). This is an easy task for a human (see figure 7), but a challenging one for a computer. Figure 7 shows that the MSSR detector is able to pick all salient natural markings.

The 3 detectors have been evaluated on a humpback whale database of 340 images (150 individuals). All results are available at biogrid.project.cwi.nl/projects/COA_gallery/scripts/coa_gallery_individuals.php. Typical results are shown on figure 8 on detail of whale tail images. The MSSR detector is able to pick up all types of salient structures as perceived by humans, unlike IBR and MSER. The detail from the second tail illustrates an indentation which has been picked up correctly only by the MSSR.

An example result for a pair of images of the same animal is shown on figure 9. Clearly the MSSR detector shows best performance identifying the most perceptually salient markings with high region repeatability for the same animal.



Figure 7: MSSR detector on a whale tail image (detail is shown on figure 8).

5 Conclusions

In this paper we have proposed a Morphology-based Stable Saliency Regions (MSSR) detector and have presented experimental results suggesting that it is well-suited to object recognition (photo-identification) problems. MSSR achieved comparable repeatability and matching performance to existing best detectors for structured images under different transformations. While the detector is only partially affine-covariant, it shows best performance in identifying regions perceptually salient to a human observer.

6 Acknowledgements

This work was supported by project NWO 613.002.056 “Computer-assisted identification of cetaceans”. Humpback database courtesy to Judy Allen, College of the Atlantic, Maine, USA.

References

- [1] E. Rangelova and E. Pauwels, “Saliency detection and matching strategy for photo-identification of humpback whales,” *ICGST International Journal on Graphics, Vision and Image Processing*, vol. Special Issue on Features and Analysis, 2006.
- [2] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool, “A comparison of affine region detectors,” *International Journal of Computer Vision*, vol. 65, no. 1/2, pp. 43–72, 2005.

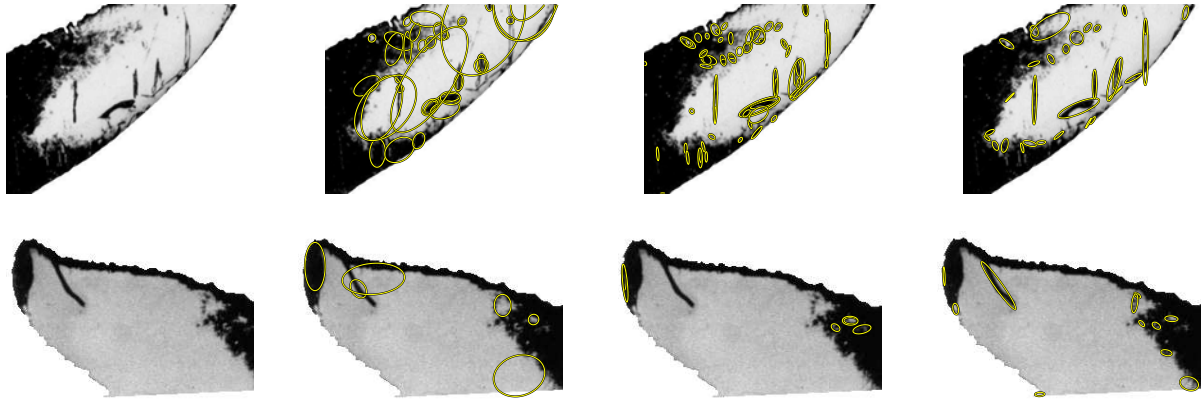


Figure 8: Region detectors on whale tails (detail). Left to right columns: original image, IBR, MSER, MSSR. Note the long indentation on the second tail missed by all, but the MSSR.

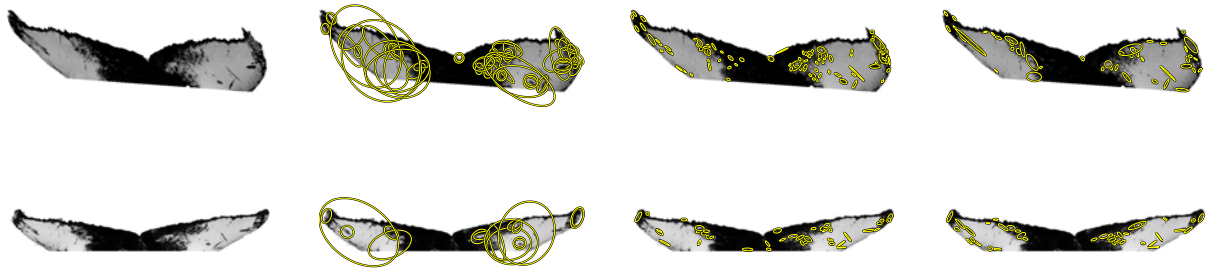


Figure 9: Region detectors for a pair of whale images. Left to right columns: original image, IBR, MSER, MMSR.

- [3] T. Tuytelaars and L. V. Gool, "Content-based image retrieval based on local affinity invariant regions," in *3rd International Conference on Visual Information Systems, Visual 99*, pp. 453–500, 1999.
- [4] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *British Machine Vision Conference*, pp. 384–393, 2002.
- [5] T. Tuytelaars and L. V. Gool, "Matching widely separated views based on affine invariant regions," *International Journal of Computer Vision*, vol. 59, no. 1, pp. 61–85, 2004.
- [6] J. Sivic, F. Schaffalitzky, and A. Zisserman, "Efficient object retrieval from videos," in *Proceedings of the 12th European Signal Processing Conference (EUSIPCO '04)*, Vienna, Austria, Sept. 2004.
- [7] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63–86, 2004.
- [8] T. Kadir, A. Zisserman, and M. Brady, "An affine invariant salient region detector," in *European Conference on Computer Vision*, pp. 404–416, 2004.
- [9] P. Soille, *Morphological Image Analysis*. Springer, 2003.
- [10] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.
- [11] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. of the International Conference on Computer Vision ICCV, Corfu*, pp. 1150–1157, 1999.

A study of 3rd and 4th order Tikhonov smoothing term influence on the convergence of active contours

Moqing Zhang¹ and Patrice Delmas

Department of Computer Science, Tamaki Campus, the University of Auckland.

¹Email: mzha053@cs.auckland.ac.nz

Abstract

An explicit snake is a smooth closed curve which deforms towards the desired features in an image. There are two types of force controlling the motion of the snake: internal and external forces. The former constrains the snake, while the latter generates attraction forces. The currently existing snake models have generally used the same internal forces, i.e., the first and second Tikhonov smoothness force terms. To investigate the possible role of higher Tikhonov constraint parameters, third and fourth force terms are added in this study. The related theoretical equations are derived and the respective influence of the four internal force terms are examined and followed on test images. While still at the preliminary stage, the present study shows that the added internal force terms may improve the smoothness and convergence of the snake.

Keywords: active contour, contour extraction, Tikhonov

1 Introduction

An explicit snake model provides a unified solution to a set of visual problems which were treated in different ways in the past. In this model, edges, lines and object contours can be extracted by the same mechanisms. Tracking these features in videos and matching them in stereo vision system can also be realized by using the same framework. It is a powerful tool for high-level image processing.

Since the first active contour, also called snake for its characteristic motion over time, was proposed by Kass, Witkin and Terzopoulos[1] in the late 1980's, a significant number of studies has been conducted to improve and to solve the problems related to its optimal convergence. Cohen's[2] balloon model gives the snake an additional force to make the snake inflate or deflate. The GVF model proposed by Chenyang and Prince[3] computes the external force as a diffusion of the gradient vectors of a gray-level or binary edge map derived from the image. In 1993, Cohen L.D. and Cohen I.[4] suggested using Chamfer distance to edge points as external force. The basic ideas of these solutions are to increase the capture range of the external force, so that the initial snakes do not necessarily lie very close to the regions of interests.

Almost all of the existing explicit snake models have the same internal force which is composed of the first and second Tikhonov smooth force terms. Higher order Tikhonov smoothness force terms

could also have some effects on improvement of the smoothness but this was never investigated. To explore these possible effects, the third and fourth Tikhonov smooth force terms are added to the snake internal constraint in this study. After a theoretical characterisation of these added terms, the roles of the internal force terms are examined.

2 Explicit snake

2.1 Model

An explicit snake (parametric snake) is a specific type of deformable model, which is a mapping:

$$\begin{aligned}\Omega &= [0, 1] \rightarrow R^2 \\ s &\mapsto v(s) = (x(s), y(s))\end{aligned}$$

Where s denotes the curvilinear abscissa and (x, y) the Cartesian coordinates of the snake points. An explicit snake model is defined as a space of admissible deformations A and a functional E to minimize. This functional represents the energy of the model which will be minimized and has the following form:

$$\begin{aligned}E &: A \rightarrow R \\ v &\mapsto E_{snake}(v) = \int_0^1 E_{snake}(v(s)) ds \\ &= \int_0^1 E_{int}(v(s)) + E_{ext}(v(s)) ds\end{aligned}\quad (1)$$

where

$$E_{\text{int}} = \alpha(s)|v_s(s)|^2 + \beta(s)|v_{ss}(s)|^2 + T(s)|v_{sss}(s)|^2 + F(s)|v_{ssss}(s)|^2 \quad (2)$$

$$E_{\text{ext}} = E_{\text{image}}(v(s)) + E_{\text{constrain}}(v(s))$$

Assume v is a local minimum for E , equation (1) leads to the following associated Euler-Lagrange equation:

$$-(\alpha v_s)_s + (\beta v_{ss})_{ss} - (T v_{sss})_{sss} + (F v_{ssss})_{ssss} + \nabla E_{\text{image}}(v) + \nabla E_{\text{constrain}}(v) = 0 \quad (3)$$

($v(0), v_s(0), v(1)$ and $v_{ss}(1)$ are known.)

Here, $v_s(s)$, $v_{ss}(s)$, $v_{sss}(s)$ and $v_{ssss}(s)$ denote derivatives of $v(s)$, $\alpha(s)$, $\beta(s)$, $T(s)$ and $F(s)$ are the weights of $v_s(s)$, $v_{ss}(s)$, $v_{sss}(s)$ and $v_{ssss}(s)$ respectively, one can control the importance of $v_s(s)$, $v_{ss}(s)$, $v_{sss}(s)$ and $v_{ssss}(s)$ by adjusting the weights $\alpha(s)$, $\beta(s)$, $T(s)$ and $F(s)$. E_{image} refers to the image energy which correspond to the desired attributes and $E_{\text{constrain}}$ is the external constraint force. In practice, we always give a weight to the image force and external force respectively, thus equation (3) becomes:

$$-(\alpha v_s)_s + (\beta v_{ss})_{ss} - (T v_{sss})_{sss} + (F v_{ssss})_{ssss} + \kappa \nabla E_{\text{image}}(v) + \kappa_p \nabla E_{\text{constrain}}(v) = 0 \quad (4)$$

A solution can be seen either as realizing the equilibrium of the forces in the equation (4) or reaching the minimum of the energy (1).

2.2 Numerical solution

Assume $f(v) = \kappa \nabla E_{\text{image}}(v) + \kappa_p \nabla E_{\text{constrain}}(v)$, then (4) becomes:

$$-(\alpha v_s)_s + (\beta v_{ss})_{ss} - (T v_{sss})_{sss} + (F v_{ssss})_{ssss} + f(v) = 0 \quad (5)$$

Using the finite difference method approximate the derivatives of v , assume the special distance is equal to 1 constantly, then the left terms of (5) can be expressed as:

$$(\alpha v_s)_s = +\alpha_{i+1}(v_{i+1} - v_i) - \alpha_i(v_i - v_{i-1}) = +\alpha(v_{i+1} - 2v_i + v_{i-1}) \text{ for } \alpha \text{ constant}$$

$$(\beta v_{ss})_{ss} = +\beta_{i+1}(v_{i+2} + v_i - 2v_{i+1}) - 2\beta_i(v_{i+1} + v_{i-1} - 2v_i) + \beta_{i-1}(v_{i-2} + v_i - 2v_{i-1}) = +\beta(v_{i-2} - 4v_{i-1} + 6v_i - 4v_{i+1} + v_{i+2}) \text{ for } \beta \text{ constant}$$

$$(T v_{sss})_{sss} = +T_{i+2}(v_{i+3} - 3v_{i+2} + 3v_{i+1} - v_i) - 3T_{i+1}(v_{i+2} - 3v_{i+1} + 3v_i - v_{i-1}) + 3T_i(v_{i+1} - 3v_i + 3v_{i-1} - v_{i-2}) - T_{i-1}(v_i - 3v_{i-1} + 3v_{i-2} - v_{i-3}) = T(v_{i-3} - 6v_{i-2} + 15v_{i-1} - 20v_i + 15v_{i+1} - 6v_{i+2} + v_{i+3}) \text{ for } T \text{ constant}$$

$$(F v_{ssss})_{ssss} = +F_{i+2}(v_{i+4} - 4v_{i+3} + 6v_{i+2} - 4v_{i+1} + v_i) - 4F_{i+1}(v_{i+3} - 4v_{i+2} + 6v_{i+1} - 4v_i + v_{i-1}) + 6F_i(v_{i+2} - 4v_{i+1} + 6v_i - 4v_{i-1} + v_{i-2}) - 4F_{i-1}(v_{i+1} - 4v_i + 6v_{i-1} - 4v_{i-2} + v_{i-3}) + F_{i-2}(v_i - 4v_{i-1} + 6v_{i-2} - 4v_{i-3} + v_{i-4}) = F(v_{i-4} - 8v_{i-3} + 28v_{i-2} - 56v_{i-1} + 70v_i - 56v_{i+1} + 28v_{i+2} - 8v_{i+3} + v_{i+4}) \text{ for } F \text{ constant}$$

Thus (5) can be written in matrix form:

$$AV + f = 0$$

Where A is a quasi nona-diagonal circulant Toeplitz matrix:

$$A = \begin{pmatrix} a_5 & a_4 & a_3 & a_2 & a_1 & 0 & 0 & \dots & \dots & 0 & a_9 & a_8 & a_7 & a_6 \\ a_6 & a_5 & a_4 & a_3 & a_2 & a_1 & 0 & \dots & \dots & \dots & a_9 & a_8 & a_7 \\ a_7 & \dots & \dots & \dots & \dots & a_1 & 0 & \dots & \dots & \dots & a_9 & a_8 \\ a_8 & \dots & \dots & \dots & \dots & \dots & a_1 & 0 & \dots & \dots & \dots & a_1 \\ a_9 & a_8 & \dots & \dots & \dots & \dots & \dots & a_1 & 0 & \dots & \dots & 0 \\ 0 & a_9 & a_8 & \dots & \dots & \dots & \dots & a_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_9 & a_8 & \dots & \dots & \dots & a_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_9 & a_8 & a_7 & a_6 & a_5 & a_4 & a_3 & a_2 & a_1 & 0 & 0 & 0 \\ 0 & \dots & \dots & 0 & a_9 & a_8 & \dots & \dots & \dots & \dots & a_1 & 0 & 0 \\ 0 & \dots & \dots & \dots & 0 & a_9 & a_8 & a_7 & a_6 & a_5 & a_4 & a_3 & a_2 & a_1 \\ \vdots & & & & \vdots & & & & \vdots & & & & \vdots \\ \vdots & & & & \vdots & & & & \vdots & & & & \vdots \\ a_1 & 0 & \dots & \dots & \dots & 0 & a_9 & a_8 & a_7 & a_6 & a_5 & a_4 & a_3 & a_2 \\ a_2 & a_1 & 0 & \dots & \dots & \dots & 0 & a_9 & \dots & \dots & \dots & \dots & a_3 \\ a_3 & a_2 & a_1 & 0 & \dots & \dots & \dots & 0 & a_9 & \dots & \dots & \dots & a_4 \\ a_4 & a_3 & a_2 & a_1 & 0 & \dots & \dots & \dots & 0 & a_9 & a_8 & a_7 & a_6 & a_5 \end{pmatrix}$$

The nine a_i weights are derived from the above equations:

$$a = \begin{pmatrix} F \\ -T - 8F \\ \beta + 6T + 28F \\ -\alpha - 4\beta - 15T - 56F \\ 2\alpha + 6\beta + 20T + 70F \\ -\alpha - 4\beta - 15T - 56F \\ \beta + 6T + 28F \\ -T - 8F \\ F \end{pmatrix}$$

V and f denote the vector of the locii and forces of the snake points.

As explained in[1], to solve equation 5, the right-hand side of the equation is set equal to the product of a time step size and the negative time derivatives of the left-hand sides. For simplicity, assume f is constant during a time step, leading to an explicit Euler method with respect to the external force. Because the matrix A completely specified the internal forces, we can evaluate the time derivative at time t rather than time $t-1$ and consequently arrive

at an implicit Euler step for the internal forces. The resulting equation is:

$$AV_t + f_{t-1} = -\gamma(V_t - V_{t-1}) \quad (6)$$

Equation (6) can be solved by matrix inversion:

$$V_t = (A + \gamma I)^{-1}(V_{t-1} - f_{t-1}) \quad (7)$$

Note that the matrix of equation (7) needs to be inverted once if the smoothness parameters are set constant through the snake temporal evolution. This can be achieved via a LU decomposition scheme in $O(n)$ time[5][6] or through the direct computation of its coefficients[7].

3 Weights of internal forces

In equation (4) each term appears as a force applied to the snake. The first four terms are the internal forces namely the first, second, third and fourth order Tikhonov smoothness force terms where $\alpha(s)$, $\beta(s)$, $T(s)$ and $F(s)$ are their associated weights.

In this section, four groups of experiments are conducted, each group focusing on one of the Tikhonov smoothing parameter. Each parameter influence is studied over an order of magnitude. The external parameter kappa is set to 0. The initial snake is a square with width equal to 40 and the distance between the snake points are set to 1. The set of parameters values studies is shown in Table1 and the relevant results are discussed in each subsection (Figure 1, Figure 2 and Figure 3).

3.1 The first group

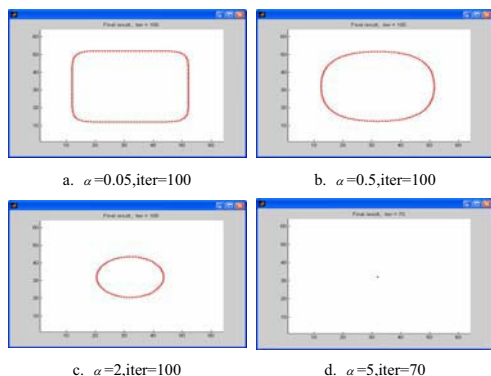


Figure 1: Snake evolution in absence of image features; Parameters values are set as in the first row of Table 1

From this group of experiments, it can be seen that a slight change in α can bring big change in topology of the snake. In the last experiment, when α

= 5, after 70 iteration, the square shaped snake becomes a point, which illustrates the elasticity imposed to the snake by α .

3.2 The second group

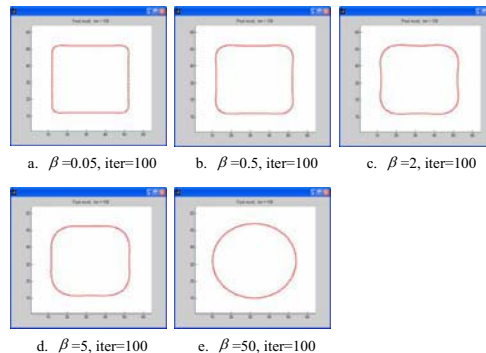


Figure 2: Snake evolution in absence of image features; Parameters values are set as in the second row of Table 1.

Compare to the first group, the affect of β is not as powerful as α . When $\beta = 5$, after 100 iteration, the snake is still in square shape, except that the four corners of the rectangle become curve. Furthermore, when $\beta = 50$, after 100 iteration, the snake turns into a circle which illustrates the smooth function of the internal force. This in line with the known curvature constraint effect of 2^{nd} order Tikhonov parameter while the first order term binds the snake elasticity.

3.3 The third group

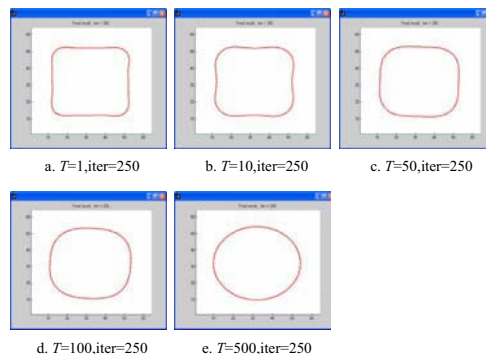


Figure 3: Snake evolution in absence of image features; Parameters values are set as in the third row of Table 1.

The snake shapes in this group are similar to those in the second group. However to gain the similar shapes the value of T in this group has to be set much higher than corresponding value in the second group. The effect of the third order Tikhonov smoothing parameter could be the same as the second order Tikhonov smoothing parameter provide that T value is set around 10 times as the value of β (refer to the corresponding Experiment e).

Table 1: The Tikhonov smoothing terms parameters values set applied in our experiment.

The focused Tikhonov smoothing parameter	parameters
1. α (first order)	a. $\alpha = 0.05, \beta = 0, T = 0, F = 0, iter = 100$ b. $\alpha = 0.5, \beta = 0, T = 0, F = 0, iter = 100$ c. $\alpha = 2, \beta = 0, T = 0, F = 0, iter = 100$ d. $\alpha = 5, \beta = 0, T = 0, F = 0, iter = 70$
2. β (second order)	a. $\alpha = 0, \beta = 0.05, T = 0, F = 0, iter = 100$ b. $\alpha = 0, \beta = 0.5, T = 0, F = 0, iter = 100$ c. $\alpha = 0, \beta = 2, T = 0, F = 0, iter = 100$ d. $\alpha = 0, \beta = 5, T = 0, F = 0, iter = 100$ e. $\alpha = 0, \beta = 50, T = 0, F = 0, iter = 100$
3. T (third order)	a. $\alpha = 0, \beta = 0, T = 1, F = 0, iter = 250$ b. $\alpha = 0, \beta = 0, T = 10, F = 0, iter = 250$ c. $\alpha = 0, \beta = 0, T = 50, F = 0, iter = 250$ d. $\alpha = 0, \beta = 0, T = 100, F = 0, iter = 250$ e. $\alpha = 0, \beta = 0, T = 500, F = 0, iter = 250$
4. F (fourth order)	a. $\alpha = 0, \beta = 0, T = 0, F = 1, iter = 250$ b. $\alpha = 0, \beta = 0, T = 0, F = 50, iter = 250$ c. $\alpha = 0, \beta = 0, T = 0, F = 100, iter = 250$ d. $\alpha = 0, \beta = 0, T = 0, F = 500, iter = 250$ e. $\alpha = 0, \beta = 0, T = 0, F = 5000, iter = 250$

3.4 The fourth group

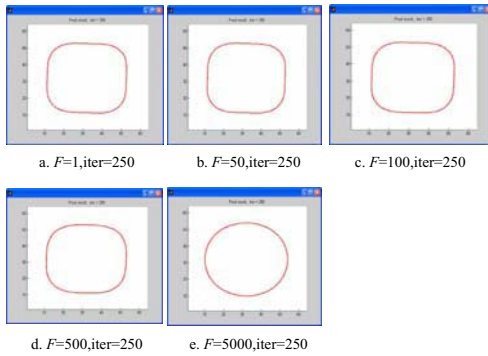


Figure 4: Snake evolution in absence of image features; Parameters values are set as in the fourth row of Table 1.

In this group, the iter values are equal to those in the third group. To achieve the similar final result, F value has to be around 10 times of T value.

The results from these experiments suggest that the effects of these internal forces can be generally categorized into two groups: impose the elasticity of the snake, while the other three (β , T and F) impose the rigidity of the snake. To achieve the similar final result, the values of β , T and F has to be set in an increasing order, this indicates that the effects of the forces controlled by these parameters are in a decreasing order, in other words, the curvature of the corresponding force impose to the snake is also in a decreasing order. Because of this, the higher order smoothness force could be

seen as a way to micro adjust the topology of the snake.

4 Test on images

In order to examine the function of the high order smoothness parameters, two groups of experiments are conducted, using different images. The first group use a 64×64 U shape binary image[8], the snake is initialized as a square (Figure 5), the distance between the snake points is 2, the weight of the image force is set to 1. Another group uses a 128×128 synthetic lip colour image, manual initialization, the distance between the snake points is to 3, the weight of the image force is set to 0.5;. In both groups, the time step is 1 and the image force is adapt from GVF model.

In each group, different values of the smoothness parameters are applied to the image. The parameters used in both groups are shown in Table 2 and Table 3 respectively.

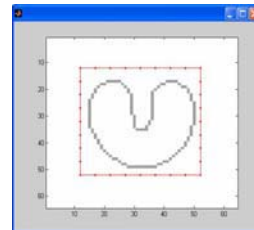


Figure 5: The U shape image (gray) and the initial snake (red).

Table 2: Parameters used in the group using the U shape binary image.

1	$\alpha = 0.01, \beta = 0.1, T = 0, F = 0, iter = 250$
2	$\alpha = 0.01, \beta = 0.1, T = 0.5, F = 0, iter = 250$
3	$\alpha = 0.01, \beta = 0.1, T = 0, F = 1, iter = 75$
4	$\alpha = 0.01, \beta = 0.1, T = 0.1, F = 1, iter = 75$

Table 3: Parameters used in the group using the lip image.

1	$\alpha = 0.1, \beta = 0, T = 0, F = 0, iter = 250$
2	$\alpha = 0.1, \beta = 0.2, T = 0, F = 0, iter = 250$
3	$\alpha = 0.1, \beta = 0.2, T = 1, F =, iter = 75$
4	$\alpha = 0.1, \beta = 0.2, T = 1, F = 1, iter = 75$

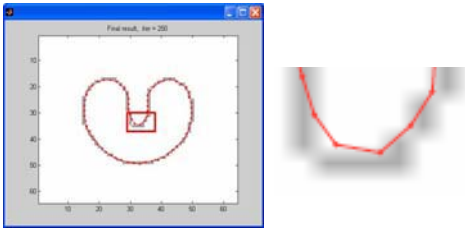


Figure 6: Left: the final result when using the parameters in the first row of Table2; Right: the zoom in of the left square area.

Figure 6 shows the result when just using α and β , the snake can not properly converge to the deep concave regions of the image feature.

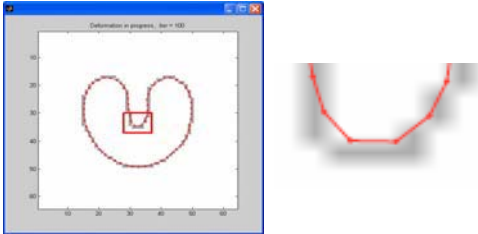


Figure 7: Left: the final result when using the parameters in the second row of Table2; Right: the zoom in of the left square area.

Once the third term joins, the final snake better converges towards the concave region, as shown in Figure 7.

From Figure 8, it can be seen that the fourth order smoothing term can play a similar role as the third term did.

Figure 9 shows that the third and fourth order smoothing applied together may achieve a better convergence.

The results corresponding to the parameters listed in Table 3 are shown in Figure 10. The left column of Figure 10 is the final results. Two sections of the lip contours are zoomed in right column. One is the

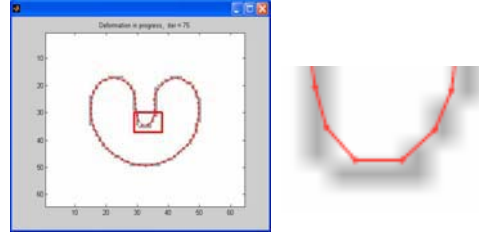


Figure 8: Left: the final result when using the parameters in the third row of Table2; Right: the zoom in of the left square area.

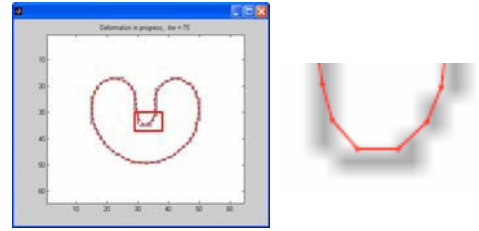


Figure 9: Left: the final result when using the parameters in the fourth row of Table2; Right: the zoom in of the left square area.

lower part of the Cupidon arch (up cell), the other is the mouth left corner (bottom cell).

According to the topology of the lip image, the extracted contour should be symmetric, the line between the two middle points of the final snake should parallel to the image edge; Regarding the left corner of the lip, because it link the up lip and the bottom lip, the snake in this area should be smoothed. It can be seen in figure 10 that the active contour achieves a better convergence when the higher smoothness terms are added.

From the experiments in this group, it can be observed that the third and fourth order parameters can help achieve a better convergence. Both of them can be used through weights T and F control micro adjustment of the smoothness and convergence.

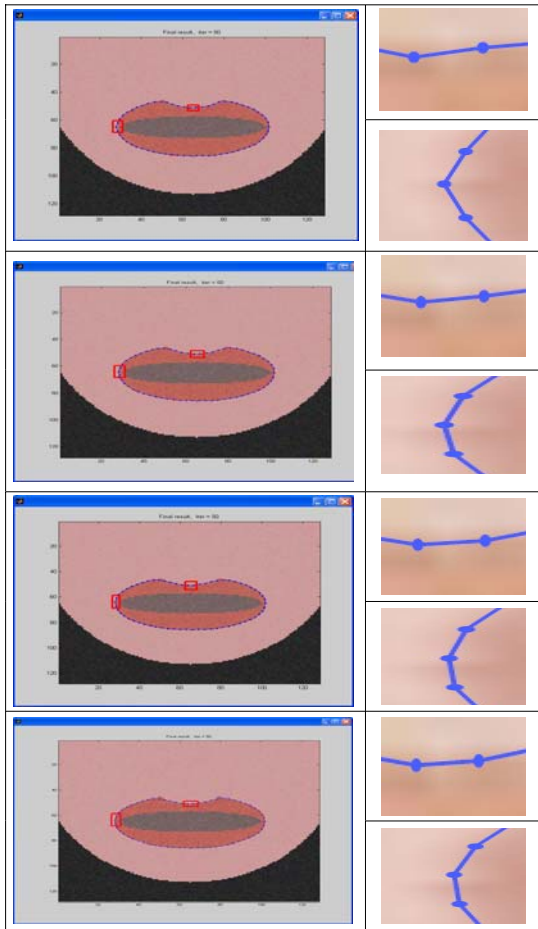


Figure 10: Left: the final results when using the parameters in the corresponding rows of Table 3; Right: top: the magnified Cupidon arch region; bottom: the magnified left corner area of the lip.

5 Conclusion

We have seen that the four smoothing terms control the snake in different way: the first order term imposes the elasticity to the snake, all the others including the second, the third and the fourth order terms impose the curvature of the snake. Because the third and the fourth order terms may provide micro-adjustment of the curvature control they may play an important role in optimal convergence of the active contour.

The present study has some implications. Theoretically a variety of higher order Tikhonov smooth terms could be developed to improve the snake control. In practice various internal forces could be chosen to improve the convergence depending on the topology of the desired feature and the image features characteristics. We are currently conducting studies to derive an optimal selection of Tikhonov smoothing terms weights.

References

- [1] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1987.
- [2] L. D. Cohen, "On active contour models and balloons," *CVGIP: Image Understanding*, vol. 53, no. 2, pp. 211–218, 1991.
- [3] C. Xu and J. L. Prince, "Gradient vector flow: a new external force for snakes," 1997.
- [4] L. D. Cohen and I. Cohen, "Finite-element methods for active contour models and balloons for 2-d and 3-d images," *Pattern Analysis and Machine Intelligence. IEEE Transactions*, vol. 15, no. 11, pp. 1131–1147, 1993.
- [5] I. Gladwell and R. Wait, eds., *A survey of numerical methods for partial differential equations*. Clarendon: Oxford, 1979.
- [6] A. Benson and D. Evans, "Mathematical software," *ACM TRANS*, vol. 3, pp. 96–103, 1977.
- [7] P. Delmas, N. Eveno, and P. Y. Coulon, "Towards automatic lip tracking," (Dunedin), Proceedings of the Image and Vision Computing New Zealand Conference, 26-28 November 2001.
- [8] C. Xu and J. L. Prince, "Gvf snake demo." <http://iacl.ece.jhu.edu/projects/gvf/snakedemo/>, visited on 1/7/2006.

Iterative Target Calibration Using Conformal Geometric Algebra

Robert J. Valkenburg¹, Nawar S. Alwesh¹, Yilan Zhao², Reinhard Klette²

¹ Industrial Research Limited, P.O. Box 2225, Auckland, New Zealand.

² Computer Science Department, The University of Auckland, New Zealand

Email: r.valkenburg@irl.cri.nz

Abstract

This paper is about real-time refinement of the 3D positions of a large number of stationary point-targets from a sequence of 2D images which are taken by a hand-held, calibrated camera group moving along an arbitrary path. To cope with the large data quantity arriving rapidly, an efficient iterative algorithm was developed. The problem and solution are expressed entirely within the computational framework of conformal geometric algebra. Experiments are performed to evaluate the algorithm based on simulated and real data.

Keywords: conformal geometric algebra, pose estimation

1 Introduction

Recovering the positions of many point-targets over a large area is computationally expensive. This paper describes an efficient iterative algorithm to refine target positions from a sequence of 2D images. The targets used are point-lights as shown in Figure 1. A group of rigidly co-located calibrated cameras, as shown in Figure 2, is moved along an arbitrary path and takes images of the targets. The image points of the targets are transformed to 3D lines which are used by the algorithm to update the 3D positions of the targets. The targets together with the camera group form part of a 6D positioning system.

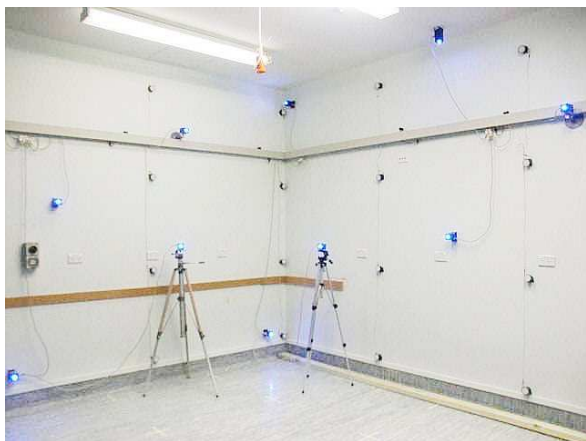


Figure 1: Targets in the laboratory.

The algorithm is expressed entirely within the computational framework of conformal geometric algebra (CGA). The previously developed target cali-



Figure 2: Camera group.

bration algorithm described in [1] is non-iterative and requires all the line data to be gathered before the algorithm can proceed. It can be used to obtain an initial estimate of the target positions which can be refined using the iterative algorithm described in this paper. This work is a continuation of work reported in [1] in the application of the conformal model of geometric algebra.

1.1 Geometric Algebra and Conformal Model

In this section, the basic concepts and operations of geometric algebra that are required in this paper are briefly introduced. For a detailed introduction to geometric algebra, refer elsewhere e.g. [2, 3, 4, 5, 6].

Geometric algebra (GA) is the application of Clifford algebras to geometric problems. It integrates many concepts and techniques, such as linear algebra, vector calculus, differential geometry, complex numbers and quaternions, into a coherent framework. A geometric algebra over \mathbb{R} is denoted $\mathcal{G}_{p,q}$ with p positive and q negative basis elements. Let x_1, x_2, \dots, x_r be vectors. $X = x_1 \wedge x_2 \wedge \dots \wedge x_r$ is referred to as an r -blade where ' \wedge ' is called *outer product*. r is the *grade* which indicates the dimensionality of the blade. A linear combination of multiple r -blades constructs an r -vector. $\mathcal{G}_{p,q}^r$ denotes the r -vectors in $\mathcal{G}_{p,q}$. A linear combination of a set of elements with different grades is a *multivector*. For example, if A is a multivector then it can be written as $A = \sum_r \langle A \rangle_r$ where $\langle A \rangle_r$ represents the grade r part of A . $\langle A \rangle$ or $\langle A \rangle_0$ represents the scalar part of A . The part of A containing the grades in another multivector B is denoted as $\langle A \rangle_B$. $A \rfloor B = \sum_{r,s} \langle \langle A \rangle_r \langle B \rangle_s \rangle_{s-r}$ is defined as the left contract inner product of A and B . The outer product can be related with the inner product by the following equation: $A \rfloor (B \rfloor C) = (A \wedge B) \rfloor C$. *Reverse* of X is defined as $\tilde{X} = x_r \wedge \dots \wedge x_2 \wedge x_1$. The dual of a blade X is defined as $X^* = X \rfloor I^{-1}$, where the pseudo-scalar I is an $(p+q)$ -blade of unit norm. The norm of a multivector A can be calculated by $|A| = \sqrt{|\langle \tilde{A} A \rangle|}$. If S is a linear operator, the *outermorphism* \underline{S} is defined by $\underline{S}(X) = S(x_1) \wedge S(x_2) \dots \wedge S(x_r)$. The derivative of multivector valued function F with respect to multivector X is denoted $\partial_X F$. $\partial_X F \hat{G}$ means differentiate $G = G(X)$ with respect to X while regarding F as a constant. The following result [7] is required in later developments,

$$\partial_X \langle XYX^{-1}Z \rangle = \langle YX^{-1}Z \rangle_X - \langle X^{-1}Z\tilde{X}YX^{-1} \rangle_X \quad (1)$$

where X, Y, Z be multivectors where Y and Z are independent of X .

GA expresses a number of models of 3D Euclidean space (\mathcal{E}^3), such as 3D Euclidean model, 4D homogeneous model and 5D conformal model. In this paper we use the conformal model of geometric algebra (CGA) based on $\mathcal{G}_{4,1}$. $\mathcal{G}_{4,1}$ is based on the orthonormal basis $\{e_1, e_2, e_3, e_+, e_-\}$ where $e_k^2 = e_+^2 = 1$ and $e_-^2 = -1$. It is usually more convenient to use the basis $\{e_o, e_1, e_2, e_3, e\}$ as it has a better geometric interpretation, where $e_o = \frac{e_- - e_+}{2}$ is associated with the origin and $e = e_- + e_+$ with the point at infinity. CGA allows a diversity of objects to be represented directly as blades (e.g. point, line, plane, circle, sphere, tangent and orientation) and allows a variety of operations to be represented as versors (e.g. rotor, translator, motor). A vector is represented as $v = v_1 e_1 + v_2 e_2 + v_3 e_3$ where

v_1, v_2, v_3 are scalars. A point with location at the Euclidean point $\vec{p} \in \mathcal{G}_3^1$ is represented as $p = \vec{p} + e_o + \frac{1}{2}\vec{p}^2 e \in \mathcal{G}_{4,1}^1$. A line is represented by $\Lambda = p \wedge v \wedge e$ where $p \in \mathcal{G}_{4,1}^1$ is a point on the line and $v \in \mathcal{G}_3^1$ is a direction vector. A line is normalised by the mapping $\Lambda \rightarrow \frac{\Lambda}{\|\Lambda\|}$. A dual sphere centered at point p with radius ρ is given by $s = p - \frac{1}{2}\rho^2 e$. A Euclidean motion is represented by a *motor* $M = \exp(-\frac{1}{2}B)$, $B = \mathbf{B} - \mathbf{t}e$, where $\mathbf{B} \in \mathcal{G}_3^2$ and $\mathbf{t} \in \mathcal{G}_3^1$. The transformation of X is by a motor M is given by $MX\tilde{M}$. A motor M has properties which are important for deriving the algorithm: (i) $M \in \text{span}\{1, e_1 e_2, e_1 e_3, e_2 e_3, e_1 e, e_2 e, e_3 e, I_3 e\} \in \mathcal{G}_{4,1}^{0,2,4}$, (ii) $MM = 1$, (iii) if $X \in \mathcal{G}_{4,1}^k$ then $MX\tilde{M} \in \mathcal{G}_{4,1}^k$.

1.2 Problem description

The N_c individual cameras which comprise the camera group are set up to approximate an omnidirectional camera, with maximum possible coverage and least amount of overlap between their image planes. Since the geometric relationships between the individual cameras are fixed and known, the camera group can be associated with a single moving coordinate system denoted by CSM . The targets are defined in a world coordinate system denoted by CSW .

An initial estimate of the positions of K targets $\{q_k^0 \in \mathcal{G}_{4,1}^1, k = 1 \dots K\}$ is given [1]. The initial pose of the camera group, CSM , is also given and represented as a motor M_o . The camera group, CSM , is moved along an arbitrary path in CSW . The movement of CSM is tracked and represented by a sequence of motors $M_n, n = 1 \dots$. At each position in CSW , N_c images of the targets are captured. Since the camera group does not provide complete coverage, some targets might not be seen whereas some might be captured in more than one image. The image points of the targets are extracted and converted to normalised lines $\{\Lambda_j \in \mathcal{G}_{4,1}^3, j \in J_{kn}\}$, in CSM , where J_{kn} represents the set of lines associated with the k^{th} target at the n^{th} position. These lines are processed to refine the initial target position estimates. When CSM is moved to the next position, the new estimate of target positions will be calculated based on the previous estimate and a new set of lines. For each position on the path an update is performed.

The problem can now be summarised as follows: Given a group of lines $\{\Lambda_j, j \in J_{kn}\}$ in CSM , a previous pose M_{n-1} , and a previous estimate of the k^{th} target q_k^{n-1} , we wish to estimate the target at the n^{th} iteration q_k^n .

2 Target refinement

The solution to the problem is developed in this section. The following steps need to be done during the target refinement for the n^{th} update: (i) update of pose of CSM , M_n ; (ii) transformation of lines $\{\Lambda_j, j \in J_n\}$ from CSM into CSW using M_n ; (iii) update of target positions in CSW , $\{q_k^n, k = 1 \dots K\}$.

2.1 Update of pose, M_n

The distance d between a point q and a line Λ is defined [7] by $d^2(q, \Lambda) = -\frac{1}{2} \langle \Lambda q \Lambda q \rangle$.

As this section deals with estimating the pose at the n^{th} position, for clarity we temporarily drop the subscript n so for example $M = M_n$. The objective function is defined as the total distance between all the points and their associated lines in CSM and is given by

$$d^2 = \sum_{k=1}^K \sum_{j \in J_k} d^2(q_k, M \Lambda_j \widetilde{M}) \quad (2)$$

where J_k represent all the lines associated with the k^{th} target (at the n^{th} position).

The pose of CSM is estimated using a Quasi-Newton non-linear minimisation technique (*Broyden-Fletcher-Goldfarb-Shanno* (BFGS) update) which is described in [8] (pages 425–430). The optimisation routine requires an objective function, its gradient and an initial estimate of the pose. The motor M representing the pose of CSM is parameterised $M = M(x)$ where $x \in \mathbb{R}^6$. We use $M(x)$ in the objective function d^2 in Equation (2) to express the objective function as $g(x) = d^2(M(x))$. The gradient is given by $[\nabla_x g(x)]_i = \partial_{x_i} g(x) = \partial_{x_i} M * \partial_M d^2$. The derivative $\partial_M d^2 = \sum_{k=1}^K \sum_{j \in J_k} \partial_M d^2(q_k, M \Lambda_j \widetilde{M})$.

For clarity, we will temporarily drop the subscripts k and j , so $d^2(q_k, M \Lambda_j \widetilde{M}) = d^2(q, M \Lambda \widetilde{M})$. Using Equation (1), and that M is a motor so $\widetilde{M} = M^{-1}$, the derivative is calculated as follows:

$$\begin{aligned} \partial_M d^2(q, M \Lambda \widetilde{M}) &= -\frac{1}{2} \partial_M \langle M \Lambda \widetilde{M} q M \Lambda \widetilde{M} q \rangle \\ &= -\langle \Lambda \widetilde{M} q M \Lambda \widetilde{M} q \rangle_M \\ &\quad + \langle \widetilde{M} q M \Lambda \widetilde{M} q M \Lambda \widetilde{M} \rangle_M \end{aligned} \quad (3)$$

The operator $\langle \dots \rangle_M$ denotes the projection of a general multivector onto the grades being present in multivector M . The optimisation returns the estimated parameters x of the motor $M(x)$.

The above update is applied at each position along the path. To estimate M_n , the initial estimate

M_{n-1} , lines $\{\Lambda_j, j \in J_{kn}\}$, and targets positions $\{q_k^{n-1}, k = 1 \dots K\}$ are passed as inputs to the optimiser. Performance improvements can be made by also passing the previous estimate of the Hessian matrix required for the BFGS update.

2.2 Update of target positions

With the estimated pose M of CSM , the given lines Λ in CSM can be transformed to CSW by $M \Lambda \widetilde{M}$. Given all the lines in CSW for all poses, the current target positions can be calculated by Lemma 1 [1],

Lemma 1 *Let $\Lambda_j \in \mathcal{G}_{4,1}^3$, $j \in J$ be a set of normalised lines and $S(x) = \sum_{j \in J} S(x, \Lambda_j)$ where $S(x, \Lambda_j) = x - (x \rfloor \Lambda_j) \rfloor \Lambda_j$. If $\underline{S} I_3 \neq 0$ then the point $q \in \mathcal{G}_{4,1}^1$ closest to all the lines in the least squares sense is given by the center of the normalised dual sphere*

$$s = -\frac{\underline{S}(I_3) \rfloor I_4}{\underline{S}(I_3) \rfloor I_3} \quad (4)$$

where $I_3 = e_1 \wedge e_2 \wedge e_3$ and $I_4 = e_o \wedge e_1 \wedge e_2 \wedge e_3$.

As the target positions are estimated in real time, an increasingly large number of lines and frequently repeated calculations would require too much computational resource. Rather than storing all the lines we update some summary variables to implement an iterative algorithm.

In Lemma 1, $\underline{S}(I_3)$ depends on all lines and vary with each update. As $\underline{S}(I_3) = S(e_1) \wedge S(e_2) \wedge S(e_3)$ it is only necessary to store and update $S(e_1)$, $S(e_2)$ and $S(e_3)$. During the iterations, the information contained in the lines needed for estimating the target positions, are accumulated in $S(e_1)$, $S(e_2)$ and $S(e_3)$. Recall S is defined as $S(x) = \sum_{j \in J} (x - (x \rfloor \Lambda_j) \rfloor \Lambda_j)$.

Let $S_{k,n}(e_i)$ denote the current estimate of $S(e_i)$ for the k^{th} target at the n^{th} iteration. $S_{k,n}(e_i)$ can be calculated based on previous $S_{k,n-1}(e_i)$, and new lines $\Lambda_j, j \in J_{kn}$ as

$$S_{k,n}(e_i) = S_{k,n-1}(e_i) + \sum_{j \in J_{kn}} (e_i - (e_i \rfloor \Lambda_j) \rfloor \Lambda_j) \quad (5)$$

From $S_{k,n}(e_i)$ the current estimate of target position is given by

$$q_k^n = s_k + \frac{1}{2} (s_k \rfloor s_k) e, \quad s_k = -\frac{\underline{S}_{k,n}(I_3) \rfloor I_4}{\underline{S}_{k,n}(I_3) \rfloor I_3} \quad (6)$$

It is not necessary to update the targets on every pose update iteration. For example, the targets may be updated after CSM has been moved by some specified distance.

3 Experiments

Experiments were carried out to test the algorithm using both simulated and real data.

The scene used in both experiments was a laboratory which is visualised in Figure 4. The room is approximately 8.7m long, 5.0m wide and 2.9m high. The targets are placed around the scene to provide reasonably even coverage shown as disks in Figure 4. Ground truth target positions ($\pm 2.5\text{mm}$) were obtained using a total station which are used in the experiments.

The camera group shown in Figure 2 was modelled and calibrated. This involves calibrating the intrinsic parameters of the cameras [9] and calibrating the pose of the cameras with respect to *CSM*.

A path for *CSM* was generated by walking for approximately 3 minutes in the laboratory and continuously acquiring the pose of the scanner head as shown in Figure 4.

3.1 Simulated Data

Lines were generated using the path by projecting the ground truth targets through the calibrated camera group model. To test the performance of the algorithm, Gaussian noise with $\sigma \in [0.1, 2.0]$ deviation pixels was added to the target image coordinates. The results are shown in Figure 3.

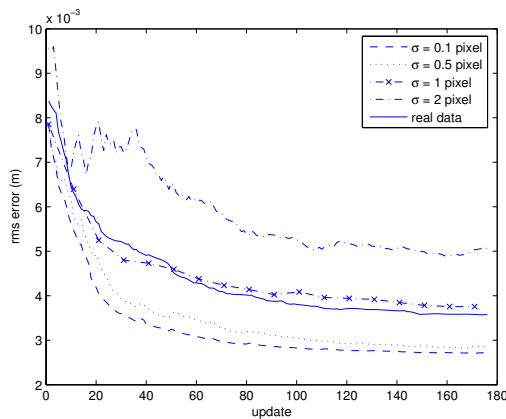


Figure 3: The *RMS* (Root Mean Square) of errors in targets vs update iterations with different levels of noise.

The results show for all noise levels there is a significant reduction in the rms error of the targets positions. With the minimum noise, the errors of estimation decrease smoothly by nearly 65%. The minimum noise of $\sigma = 0.1$ pixel is a reasonable approximation of the noise present in the real data.

The maximum noise was included to show the algorithm is stable under high noise conditions.

3.2 Real Data

The camera group was moved along the path shown in Figure 4. The algorithm was applied to the acquired line data. Results for real data are shown in Figure 3. The rms errors of estimation decrease smoothly by nearly 57%

They are not as good as the low noise level simulations due to system errors that are unrelated to the actual algorithm. For example, (i) the measurement errors of the ground truth targets play no role in the simulation results as they do for the real data, (ii) the camera group model calibration errors have no influence on the simulation results since the same model is used for both projection (targets mapped to image points) and backprojection (image points mapped to lines). With the real data, the camera model may not perfectly model the physical camera group. The algorithm can

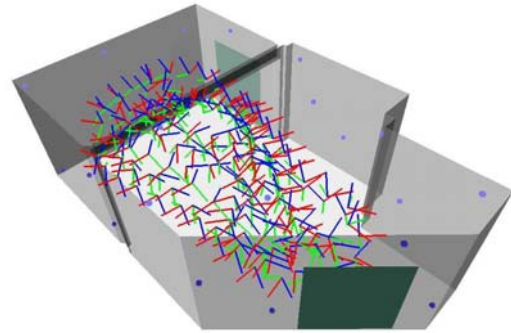


Figure 4: A model of the laboratory used. Disks are estimated targets; the figure also shows a few *CSM* coordinate systems along the path of the camera group.

update the target positions at 30Hz on a standard 3GHz PC.

4 Conclusion

We developed an iterative algorithm for refining 3D target positions over a large number of images. The data is gathered by moving a camera group along an arbitrary path in the scene. The iterative target update algorithm performed well over a wide variety of noise conditions. The algorithm has low storage and computational requirements.

The use of the conformal model of geometric algebra (CGA) benefits the development of the solution in both theory and practice. CGA provides a compact symbolic representation of objects and their transformations. A variety of objects

(e.g., vectors, points, lines, spheres) and operations (e.g. motors) can be represented in a single algebra which simplifies the implementation. The use of a single motor element to represent a Euclidean transformation (instead of separate rotation and translation), further simplified the implementation.

5 Acknowledgements

This work was supported by the New Zealand Foundation for Research, Science and Technology.

References

- [1] R. J. Valkenburg and N. S. Alwesh, "Calibration of target positions using the conformal model and geometric algebra," in *Image Vision Computing New Zealand*, pp. 241–246, Otago University, 2005.
- [2] L. Dorst and S. Mann, "Geometric algebra: a computational framework for geometrical applications (part i: algebra)," *IEEE Computer Graphics and Applications*, vol. 22, pp. 24–31, May/June 2002.
- [3] S. Mann and L. Dorst, "Geometric algebra: a computational framework for geometrical applications (part ii: applications)," *IEEE Computer Graphics and Applications*, vol. 22, pp. 58–67, July/August 2002.
- [4] E. Bayro-Corrochano and G. Sobczyk, eds., *Geometric Algebra with Applications in Science and Engineering*. Birkhäuser, 2001. Old wine in new bottles: A new algebraic framework for computational geometry, chapter 1.
- [5] D. Hestenes and G. Sobczyk, *Clifford algebra to geometric calculus*. D. Reidel Publishing Company, 1984.
- [6] D. Hestenes, *New Foundations for Classical Mechanics*. Kluwer Academic, second ed., 1999.
- [7] R. J. Valkenburg, "Some techniques in geometric algebra for computer vision," Tech. Rep. 8713000-1-03, Industrial Research Limited, Auckland, New Zealand, August 2003.
- [8] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C++*. Cambridge University Press, second ed., 2002.
- [9] R. J. Valkenburg and D. W. Penman, "Accurate unsupervised camera intrinsic calibration," in *IVCNZ*, pp. 215–219, 2004.

Fingerprint Matching using Enhanced Shape Context

Paul W.H. Kwan¹, Junbin Gao², and Yi Guo¹

¹ School of Mathematics, Statistics and Computer Science, University of New England,
Armidale NSW 2351, Australia.

² School of Information Technology, Charles Sturt University, Bathurst NSW 2795, Australia.

Email: {kwan, yguo4}@mcs.une.edu.au, jbgao@csu.edu.au

Abstract

Shape context, a robust descriptor for point pattern matching, is applied in fingerprint matching by enhancing with minutiae type and angle details. A modified matching cost between shape contexts, by including the application specific contextual information, improves the accuracy of matching when compared with the original definition. To reduce computation for practical use, a simple pre-processing step termed elliptical region filtering is applied in removing spurious minutiae prior to matching. Empirical experiments conducted on a database of fingerprint images confirmed the improvements in accuracy and speed attained by the proposed method.

Keywords: fingerprint matching, minutiae, shape context, elliptical region filtering, point pattern matching

1 Introduction

Biometric recognition is a technology for confirming a person's identity based on his physiological and behavioural traits. Among these, fingerprints, face, speech, iris and hand geometry are most commonly used [1]. An early application of this technology was Automatic Fingerprint Identification System (AFIS) found in law enforcement. Recently, a number of non-forensic applications like secured access to restricted areas, network login, etc appeared.

Post-evaluation of the terrorist attacks that occurred in America on September 11, 2001 called for increase in surveillance both within the country and at border control. Primary activities are identity verification and matching against a list of suspects. To automate these activities, America and her partners in the visa-waiver program developed their biometric passports based on standards defined by the International Civil Aviation Organisation (ICAO). Currently, digital images of fingerprints, face, or iris are stored only. The main performance requirements are accuracy and speed.

This work attempts to address both the accuracy and speed in automatic fingerprint identification. The method proposed belongs to the class of minutiae-based fingerprint matching [1]. Like most methods in this class, the presence of spurious minutiae adversely affects the accuracy. In addition, the computational overhead could increase considerably, rendering the method impractical for online application.

In practice, one can model minutiae matching as a kind of point pattern matching. Recently, a robust descriptor called shape context was proposed and its effectiveness demonstrated in general shape matching involving point patterns [2]. In the original paper,

application-specific contextual information was not considered. The sets of points on both shapes were randomly selected. However, as the authors pointed out, application-specific contextual information can be exploited to improve accuracy. This is one of the major motivations underpinning this work.

In this paper, our contributions are two-fold. First, a simple pre-processing method called *elliptical region filtering* is proposed to filter out potential spurious minutiae that were introduced in preliminary minutiae extraction. To assess its effectiveness, two simple metrics motivated by the precision and recall adopted in information retrieval research are defined and verified by experiments.

Second, by including minutiae type and angle details as application-specific contextual information, we are able to improve the matching accuracy versus speed ratio over the original shape context in cases where large number of minutiae were removed by filtering.

The rest of the paper is organized as follows. Section 2 briefly reviews related work. Section 3 explains our method in detail, supported by empirical experimental results. Section 4 concludes with future directions.

2 Related Work

Fingerprint matching methods can be largely grouped into three main classes, including correlation-based matching, ridge feature-based matching, and minutiae based matching [1]. In correlation-based matching, correlation between corresponding pixels on a pair of fingerprint images is computed for various alignments like displacement and rotation and used for matching. In ridge feature-based matching, features like local orientation, frequency and shape of ridge patterns are

used. In minutiae based matching, minutiae are first extracted from the fingerprint images and stored as sets of points on a two-dimensional plane. Matching essentially consists of finding the alignment between the template and the input minutiae sets that result in the maximum number of pairings. Fig. 1 shows a fingerprint with two kinds of minutiae marked.

The process of finding an optimal alignment between the template and the input minutiae sets can be modelled as point pattern matching. Recently, the shape context, a robust descriptor for point pattern matching was proposed in the literature. According to experiments on the MNIST handwritten digit database and the MPEG-7 shape silhouette database, the shape context was reported to outperform a number of well-known methods [2]. However, in the original definition, it only considered the distribution of the remaining points with respect to each selected point. While the authors mentioned the possibility of including application specific contextual information in the definition, no concrete suggestion was given. In this work, both minutiae type and angle details are applied as application specific contextual information to enhance the original shape context, leading to improved matching accuracy.

A major problem that could degrade the accuracy of minutiae-based matching is due to spurious minutiae extracted from poor quality fingerprint images. These could result from dry skin, a person's age or his occupation. Pre-processing steps utilizing fingerprint enhancement and minutiae filtering algorithms have been reported in the literature [3, 1]. In this work, a method similar to Hong et al. [3] was applied in fingerprint enhancement, while a method called elliptical region filtering is proposed to remove potential spurious minutiae. Spurious minutiae often occur along the edge between areas of a fingerprint that either touch or not touch the scanner surface.



Figure 1: Fingerprint with minutiae marked.

3 Proposed Method

Existing minutiae-based matching methods largely comprises the following four processing stages:

- *Fingerprint enhancement:* features in poor quality fingerprint image are enhanced before extraction.

- *Minutiae extraction:* in most cases, ridge endings and ridge bifurcations are detected and extracted from the enhanced image.
- *Minutiae filtering:* an optional stage in which spurious minutiae that can degrade both accuracy and speed of matching are filtered.
- *Fingerprint matching:* two sets of minutiae, one for the input and another for the template, are matched. A score that measures their similarity (or dissimilarity) is computed and compared to a threshold to decide either acceptance or rejection.

Here, our contributions lie in the stages of minutiae filtering and fingerprint matching. For completeness sake, we will describe briefly fingerprint enhancement and minutiae extraction as used in this work.

Fingerprint enhancement follows largely the approach reported in Hong et al. [3]. First, ridge regions in the input image are identified and normalised. Next, the ridge orientations are determined. Third, local ridge frequencies are calculated. Fourth, contextual filters with the appropriate orientations and frequencies are applied. Fifth, binarization is performed resulting in a black and white image of the fingerprint. Fig. 2's upper half presents the output of the steps in the enhancement process.

Minutiae extraction starts by thinning the black and white fingerprint image resulted from enhancement. From the thinned image, potential minutiae are detected and extracted by tracking the set of one pixel width edges. Fig. 2's lower half shows both the thinned image and the minutiae extracted, with 'o' indicates ending and '+' a bifurcation.

3.1 Minutiae Filtering

In this work, a simple method called *elliptical region filtering* is proposed to remove potential spurious minutiae in order that the computational overhead in matching a pair of fingerprints can be reduced while accuracy could be maintained. This method is motivated by the observation that the surface of a fingerprint that touches the sensor can be modelled as an ellipse (Refer to the filtered image in Fig. 2 for the idea). As a result, true minutiae are more likely extracted within the ellipse than would be on or outside the boundary (Refer to the bottom left image in Figure 2 for the argument).

As mentioned, the method we proposed makes use of a simple geometric property of an ellipse. In the ellipse shown in Fig. 3, the centre is M while both F1 and F2 are the foci that lie on its major axis. The length of the major axis is $a/2$, while that of minor axis is $b/2$. For any point P , one way to determine if it is inside an ellipse is by using the following formula:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} < 1 \quad (1)$$

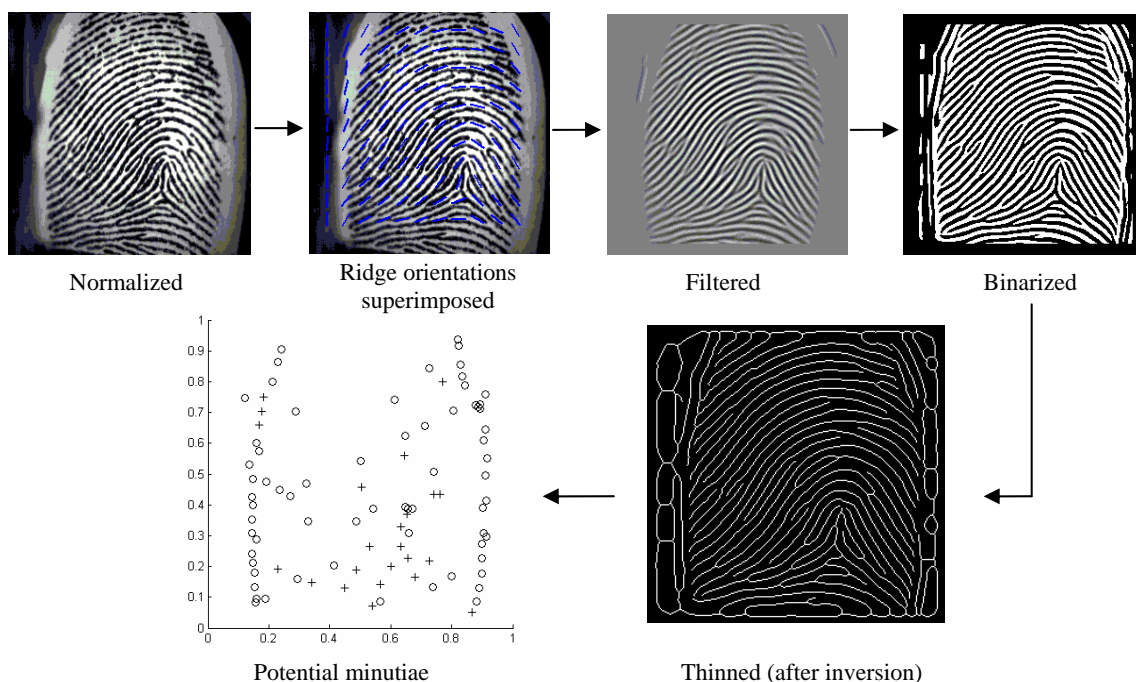


Figure 2: Fingerprint enhancement and Minutiae extraction

where (x,y) are coordinates of P on a plane. Here, our objective is to remove minutiae that lie outside of the ellipse. That is, we want to retain points like $P1$ but not $P2$ or $P3$. To compute this efficiently, we make use of another definition of ellipse that the circumference is the set of points such that the sum of the distances from each point to the two foci equals $2a$, which is the length of the major axis. In other words, a point is inside an ellipse if the sum of the distances is less than $2a$.

Formally, if we denote the two foci vectors that point from M to $F1$ and M to $F2$ by $C1$ and $C2$ respectively, a point P is inside the ellipse when

$$|P - C1| + |P - C2| < 2a \quad (2)$$

Using this idea, filtering minutiae is carried out as follow. First, we compute the smallest *rectangular bounding box* that encloses all the minutiae. Taking the longer side of the box as the length of the major axis of an ellipse, we fit it inside the bounding box. By referring to equation (2), we determine if each of the minutiae falls inside the ellipse or not. Those that are not will be filtered.

Fig. 4 shows the sets of minutiae extracted both before and after applying elliptical region filtering. The one on the left plot is the pre-filtered set and has 90 minutiae altogether, while the one on the right plot has 53 minutiae, just several more than half the size of the pre-filtered set. By comparing the before and after images, we can verify that most of the false endings due to the boundary effect have been filtered, while those that lie within the centred or “confident” region remained.

We understand that this simple filtering procedure can potentially remove genuine minutiae as neither the structural nor the contextual information of each minutia was examined. To evaluate its effectiveness empirically, we define two simple metrics based on the *precision* and *recall* that are used in Information Retrieval research as follows:

$$precision = \frac{\# \text{ genuine minutiae}}{\# \text{ extracted minutiae}} \quad (3)$$

$$recall = \frac{\# \text{ genuine minutiae}}{\# \text{ ground truth minutiae}} \quad (4)$$

Here, the set of genuine minutiae is the subset of the total extracted minutiae that are in the set of *ground truth* minutiae. By ground truth minutiae, we are referring to those that are agreed visually by 2 out of 3 human subjects. Using the same fingerprint shown in Fig. 1 and Fig. 2 as example, the number of ground truth minutiae is 37. Out of 90 minutiae in the pre-filtering set, 33 are genuine minutiae. Compared to this, out of 53 minutiae in the post-filtering set, 32 are genuine minutiae.

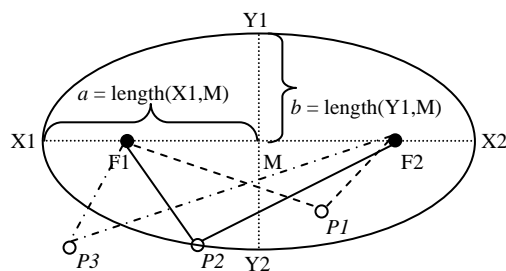


Figure 3: An ellipse showing the centre and foci

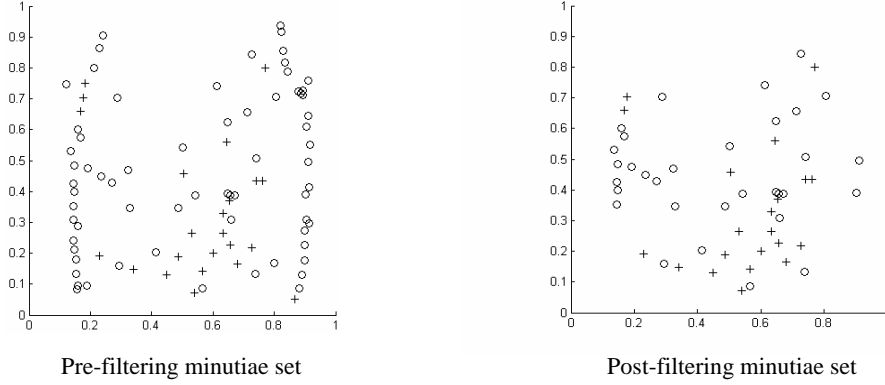


Figure 4: The sets of minutiae both before (90) and after (53) applying elliptical region filtering

Based on equations (3) and (4), the values for *precision* and *recall* are 0.36 and 0.89 for pre-filtering, while those for post-filtering are 0.60 and 0.86. By comparing these figures, we notice that *precision* in fact improves in the post-filtering set while *recall* is largely maintained. Comparable figures were obtained in our extensive experiments involving other fingerprint images in the database. Based on these empirical results, the following two conclusions can be drawn. First, the proposed filtering method does not adversely affect the set of genuine minutiae extracted. Second, both the total number and the number of false minutiae (factors that might decrease matching accuracy while increasing computational load) are largely reduced.

3.2 Fingerprint Matching

In this section, we will explain how the shape context proposed recently in [2] is enhanced and applied in matching a pair of fingerprints whose minutiae are modelled as point patterns. To provide the necessary background for our explanation, we briefly summarize below how the shape context is constructed for the set of filtered minutiae of a fingerprint shown in Fig. 5. They will be used in matching the minutiae of the fingerprint shown in right hand plot of Fig. 4 in our discussion.

Basically, there are four major steps in the shape context based fingerprint matching (Fig. 6):

- *Construct shape context:* for every minutia p_i , a coarse histogram h_i of the relative coordinates of the remaining $n - 1$ minutiae is computed,

$$h_i(k) = \#\{q \neq p_i : (q - p_i) \in \text{bin}(k)\}. \quad (5)$$

The bins are uniform in the log-polar space (Fig. 5). To measure the cost of matching two minutiae, one on each of the fingerprints, the following formula based on the χ^2 test statistic:

$$C_{ij} \equiv C(p_i, q_j) = \frac{1}{2} \sum_{k=1}^K \frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)}. \quad (6)$$

The set of all costs C_{ij} for all pairs of minutiae p_i on the first and q_j on the second fingerprint are similarly computed.

- *Minimize matching cost:* given all costs C_{ij} in the “current” iteration, this step attempts to minimize the total matching cost,

$$H(\pi) = \sum_i C(p_i, q_{\pi(i)}). \quad (7)$$

Here, π is a permutation enforcing a one-to-one correspondence between minutiae on the two fingerprints. (Top right hand plot of Fig. 6 illustrates the set of initial correspondences)

- *Warping by TPS transformation:* given the set of minutiae correspondences, this step tries to estimate a modelling transformation $T: R^2 \rightarrow R^2$ using thin plate spline (TPS) to warp one onto the other. The objective is to minimize bending energy of the TPS interpolation by $f(x,y)$ as,

$$I_f = \iint_{R^2} \left(\frac{\partial^2 f}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 f}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 f}{\partial y^2} \right)^2 dx dy \quad (8)$$

Further details on the form of the interpolant $f(x,y)$ and the interpolation conditions can be found in [2].

This and the previous two steps are repeated for several iterations (5 in our experiments) before the final distance that measures the dissimilarity of the pair of fingerprints is computed. (Refer to the two bottom plots of Fig. 6 for the idea)

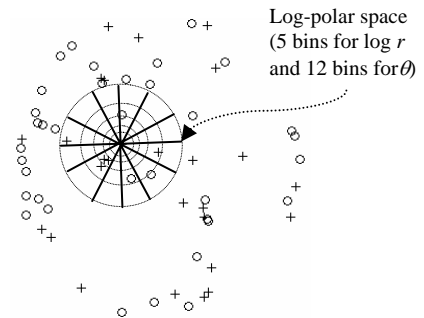


Figure 5: Shape context of a fingerprint's minutia

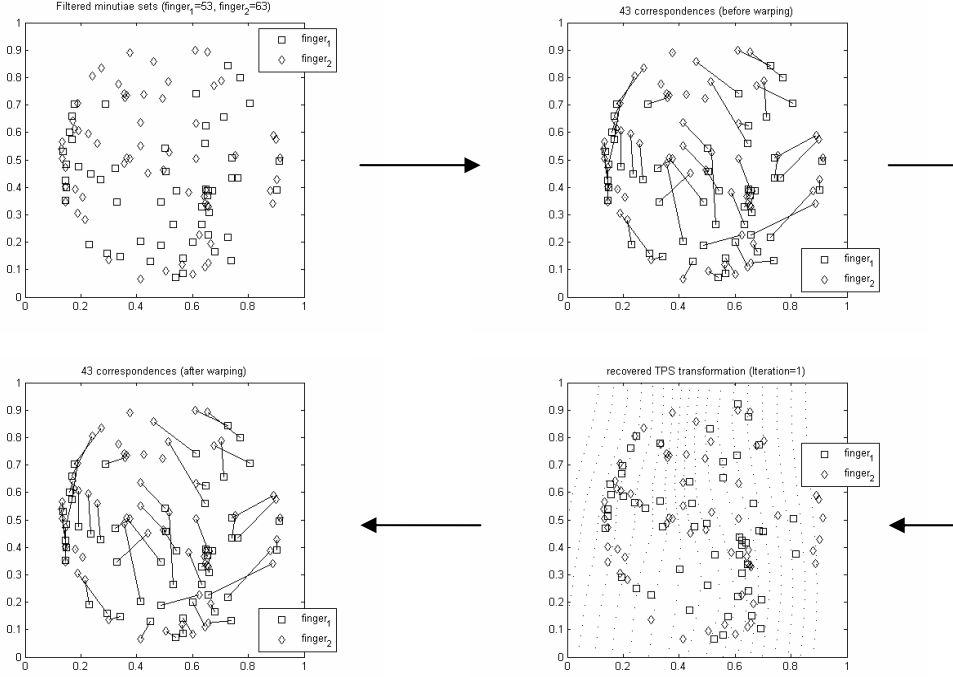


Figure 6: Major steps in shape context based fingerprint matching

- *Calculate final distance:* in the original paper, the final distance D is defined as,

$$D = D_{sc} + \alpha D_{ac} + \beta D_{be}. \quad (9)$$

where D_{sc} is the shape context cost calculated after the iterations, D_{ac} an appearance cost, and D_{be} the bending energy. Both α and β are constants determined by experiments. In this work, D_{ac} is not included as it is relevant only for grayscale images while the image here is binary. The distance D is thus defined as,

$$D = D_{sc} + \beta D_{be}. \quad (10)$$

By repeated experiments using our database, the optimal value for $\beta \in [0,1]$ is found to be 0.1. For each fingerprint in the database, its distance from the input is calculated. A final ranking in which the top has the least distance from the input is obtained.

3.2.1 Enhanced Shape Context

Initially, we applied shape context without filtering. The database we used contains 21 different fingers, each having 8 impressions totalling 168 fingerprint images. For our notation, 1_1 denotes the first impression of finger 1 while 21_8 the eighth of finger 21. Although the size of our database might be small, it is adequate for illustrating our idea.

Even in the presence of a larger number of spurious minutiae, matching by the original shape context is still quite effective (Refer to the two pre-filtered ranking columns in Table 1). For example, when input fingerprint is 1_1, all remaining 7 impressions

plus itself came in the top 8 of the final ranking. For fingerprint 2_1, though the result is not as spectacular as 1_1, 6 out of 8 impressions came in the top 9 ranking. This speaks strongly the intrinsic robustness of the shape context matching model.

But, the practical problem we face is speed. Even though CPU time cannot be considered an accurate estimate of computational load, it could provide an idea on how efficient fingerprint matching with the original shape context runs. In the case of matching 1_1 and 2_1 against the database, the times are 697 sec and 850 sec, limiting its usefulness in practice.

Table 1: Matching results by original shape context

		1_1		2_1	
	pre-filtered ranking	post-filtered ranking		pre-filtered ranking	post-filtered ranking
1_1	1	1	2_1	1	1
1_2	4	3	2_2	29	57
1_3	6	6	2_3	9	4
1_4	2	2	2_4	3	41
1_5	7	84	2_5	2	2
1_6	3	4	2_6	100	61
1_7	5	5	2_7	7	129
1_8	8	45	2_8	5	75

To reduce execution time, we apply filtering before matching. The matching times become 120 sec and 161 sec, which are less than 20% the pre-filtered figures. However, accuracy is degraded when we compare the rankings in the post-filtered columns with those of the pre-filtered columns in Table 1. The goal of improving the matching accuracy while maintaining the reduction in execution time leads to *enhanced shape context* that is described below.

In our formulation, minutiae type and angle details are incorporated as application-specific contextual information to enhance the original shape context. To accomplish that, we define a new matching cost C_{ij}^* between two minutiae p_i and q_j as,

$$C_{ij}^* \equiv C^*(p_i, q_j) = (1 - \gamma C_{ij}^{type} C_{ij}^{angle}) C_{ij} \quad (11)$$

Here, C_{ij} is the original shape context cost defined in equation (6), C_{ij}^{type} the cost in matching the type of p_i and q_j , C_{ij}^{angle} the cost in matching the ridge orientations tangent at p_i and q_j respectively, and $\gamma \in [0,1]$ whose optimal value is tuned by repeated experiments. Note that the multiplications between C_{ij}^{type} , C_{ij}^{angle} and C_{ij} in equation (11) are scalar (i.e., element-by-element) rather than the usual matrix multiplication. C_{ij}^{type} and C_{ij}^{angle} are defined as,



$$C_{ij}^{type}(p_i, q_j) = \begin{cases} -1, & \text{type}(p_i) = \text{type}(q_j) \\ 0, & \text{type}(p_i) \neq \text{type}(q_j) \end{cases} \quad (12)$$

where the *type* is either ridge ending or bifurcation.

$$C_{ij}^{angle}(p_i, q_j) = -0.5 * (1 - \cos(\angle_{initial-warped})) \quad (13)$$

where $\angle_{initial-warped}$ is the absolute value of the difference in ridge orientations tangent at p_i and q_j in the beginning and after each iterative warping. If $\angle_{initial-warped}$ is greater than π , it is adjusted as $(2\pi - \angle_{initial-warped})$ so it will be less than or equal to π .

Table 2: Compare matching results by original and enhanced shape contexts

	 1_1		 2_1	
	original shape context	enhanced shape context	original shape context	enhanced shape context
1_1	1	1	2_1	1
1_2	3	3	2_2	57
1_3	6	6	2_3	4
1_4	2	2	2_4	41
1_5	84	48	2_5	2
1_6	4	4	2_6	61
1_7	5	5	2_7	129
1_8	45	41	2_8	75

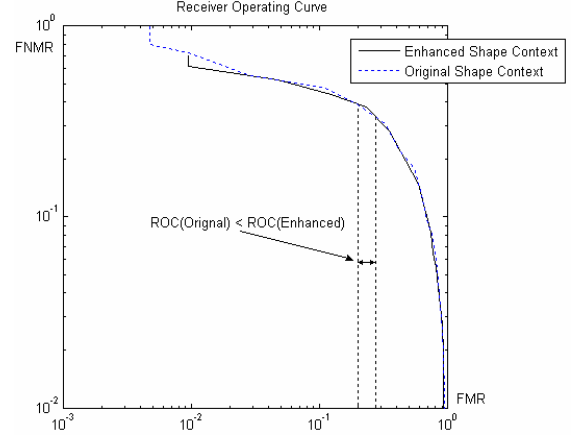


Figure 7: Compare ROCs between the enhanced and original shape contexts

In Table 2 the enhanced shape context is compared with the original after filtering is applied. For 1_1, it is clear that the enhanced shape context improves the rankings of both 1_5 and 1_8 while the other six remain same. For 2_1, most rankings improve except for 2_3. But, it is able to preserve the top 3 entries of the pre-filtered column in Table 1, which is not possible with the original shape context.

Next, in Figure 7, we compare them over the entire database by the ROCs (Receiver Operating Curves) constructed using the FMR (False Match Rate) and the FNMR (False Non-Match Rate). These figures are computed from 588 genuine and 210 imposter matching attempts as in [1]. Other than the small range indicated in the figure, the ROC of enhanced shape context consistently achieves a slightly lower or equal joint FMR and FNMR than the original.

4 Conclusions

In this paper, the shape context descriptor is applied in fingerprint matching by enhancing with minutiae type and angle details. The modified shape context cost improves matching accuracy when compared to the original definition. To reduce computation, elliptical region filtering is proposed for removing spurious minutiae prior to matching. Experiments confirmed the improvements in accuracy and speed attained by the proposed method.

5 References

- [1] D. Maltoni, D. Maio, A. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*: Springer, 2003.
- [2] S. Belongie, J. Malik, and J. Puzicha, "Shape Matching and Object Recognition Using Shape Context," *IEEE Trans. PAMI*, vol. 24, no. 24, pp. 509–522, 2002.
- [3] L.Hong, Y. Wan, and A. Jain, "Fingerprint Image Enhancement: Algorithm and Performance Evaluation," *IEEE Trans. PAMI*, vol. 20, no. 8, pp. 777–789, 1998.

Towards real time difference imaging in the far blue (390-440 nm)

G.M. Miskelly¹, J.H. Wagner¹

¹Department of Chemistry, City Campus, University of Auckland

Email: g.miskelly@auckland.ac.nz

Abstract

Improved detection of some types of forensic trace evidence has been achieved by combining narrow band images taken at and bracketing the spectral feature of the evidence of interest. This approach has been successfully applied to blood imaging where it provides improved sensitivity and selectivity towards blood and significant reduction in background patterning. We have been attempting to further develop this technique by building a portable near real time imaging system which can rapidly capture two images simultaneously, perform the necessary image processing, and display the result within a time frame that would make it practical enough to be used to search for traces of blood at a crime scene. This has required the integration of optics, cameras, lighting equipment and software. Particular attention is being paid to light throughput and the ability to image at a variety of working distances and fields of view. This paper reports on the progress towards the completion of the camera system, current hurdles and limitations, and proposed future applications.

Keywords: Forensic, Spectral imaging, Difference imaging, LEDs,

1 Introduction

Utilising spectral features of certain types of trace evidence as a means to improve their detection has gained popularity in forensic science [1], [2]. A simple (and well established) example of this is the use of blue light to improve the contrast of images of blood evidence [3]. The method can be further improved by arithmetically combining images taken at narrow wavelengths which target and bracket the narrow Soret absorption band exhibited by blood [4]. Comparable results can also be obtained by combining two images instead of three [4], although these can be subject to increased background interference. Figure 1 shows the strong absorption band present in blood and the narrow bands used in the imaging process.

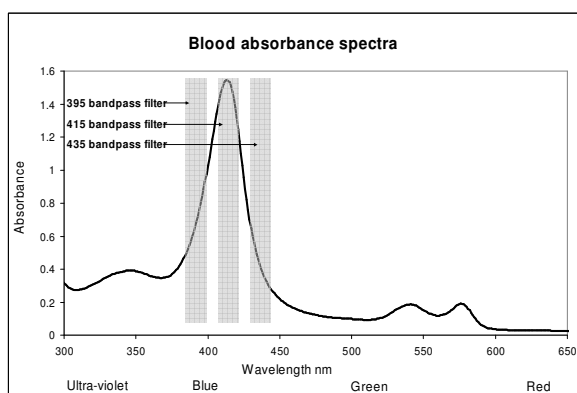


Figure 1: Absorbance spectra of blood (1000x dilution) and the spectral sampling bands.

In our current protocol for imaging blood, images are captured sequentially using a filter wheel or liquid crystal tuneable filter, LCTF to select the appropriate wavelength regions. The time required for the exposures requires that camera and subject be still relative to each other. In addition, there is also some delay between initial image capture and the display of the processed image. For these reasons our current system is impractical for rapidly locating blood evidence at crime scenes. A practical imaging system for the location of blood staining would need to be handheld, capture images simultaneously, and operate in the millisecond timeframe. We also wanted to be able to offer a similar level of flexibility to the end user that they have with a conventional camera, i.e. the ability to focus over a range of working distances and fields of view while maintaining acceptable image quality, together with short exposure times required for a hand held device. To simplify the system, the two wavelength method was chosen for development.

The simultaneous capturing of multiple images of the same scene is not new. Many professional video cameras now use three CCDs to capture the red green and blue components separately to provide improved resolution. The Dual View™ from Optical Insights [5] captures and filters two spatially identical images simultaneously and relays them side by side onto one sensor. This allows for quick processing but reduces the available sensor resolution by a factor of two and the relative throughputs for imaging at the two wavelengths must be similar because the gain and integration time is consistent across the sensor. This device has been used successfully in the fluorescence ratio microscopy imaging of thermo-responsive

neurons [6]. Astronomers have developed a camera system using dichroic mirrors and relay optics which separates an image into 15 spectral components and images each simultaneously on to 15 separate CCDs [7]. Our proposed design uses a beamsplitter to direct light towards two cameras simultaneously, eliminating the sequential capture required when using a single camera. Each lightpath would incorporate a filter at 415 nm and 435 nm respectively and the images would be processed in software in this prototype device.

Real time imaging in the blue region is made difficult by the relatively low quantum efficiency of most commercial CCD and CMOS sensors in this region. For example a Qimaging Qicam scientific camera has an approximate quantum efficiency of 20% at 400 nm. Compounding this problem is the relatively low output intensity of many standard light sources in the blue region and that the technique requires sampling a fairly narrow band of blue light which further reduces the available light to the sensor. This results in long exposure times to obtain a sufficient signal to noise ratio which is not practical for real-time imaging. To address this problem we evaluated several different types of light sources for their ability to provide a sufficient amount of blue light. Figure 2 shows the spectral distribution of five different sources of blue light, normalised to fit on the same axis.

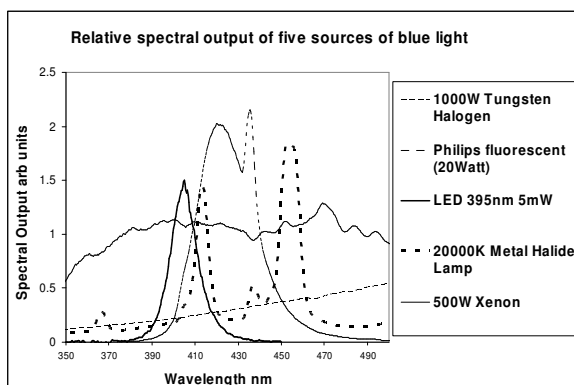


Figure 2: Relative spectral output of five sources of blue light.

The most intense blue light sources were the metal halide and xenon lamps. However both these types of lights required heavy power supplies and cooling making them impractical for our purpose. The fluorescent light provided good output at the required wavelengths and was fairly lightweight but the highly diffuse nature of output from the fluorescent tube made it difficult to focus the light. We therefore turned our attention to light emitting diodes, (LEDs), as potential light sources. LEDs are an attractive option for a number of reasons: they are very light and have low power requirements making them practical for a portable light source, they can be pulsed at higher than their nominal currents providing higher output intensity, and two different LED types can be combined in a way which allows control over

the relative spectral output of the light source. This last feature is useful because the sensitivity of most CCDs drops fairly dramatically over the range 450-400 nm which normally means the exposure times would need to be longer for the shorter wavelengths under standard lighting conditions. By using more LEDs at the lower wavelengths or by driving them at higher currents the spectral output of the LED light source can be made to counteract the difference in sensitivity of the camera sensor allowing for equal exposure times of the two cameras. LEDs emit light over a fairly narrow range, typically 20-40 nm, therefore it was necessary to find LEDs whose peak spectral output was as close to our desired wavelength as possible.

A design incorporating two CCD cameras, a single imaging lens, a beam splitter, relay optics, interference filters and a synchronised LED flash unit was settled upon. Incorporated in this way, each camera would rapidly capture an image of the same scene at different wavelengths and pass the images into a computer for processing and display.

2 Experimental

2.1 Equipment

The cameras used in the prototype were a Qicam (Qimaging Corp, 1/2" 10-bit, monochrome CCD, Firewire, with 1360x1036 active pixels) and a Retiga (Qimaging Corp, 1/2" 12-bit, monochrome CCD, Firewire, with 1360x1036 active pixels) The imaging lens was a c-mount Pentax 16mm lens. Beamsplitter (part N54-824) relay optics (part N55-272) and other integrating components were purchased from Edmund Optics, and adaptors were made to complete the integration. LEDs with a peak wavelength of 435 nm were purchased from Roithner Lasertechnik (part LED435-12-30) and LEDs with a peak wavelength of 420 nm were purchased from LEDreps (part UVA-L5N20K-xx) The image capture and processing were performed in V++ (Digital Optics) and custom electronics were designed to simultaneously trigger the two cameras and the LED flash unit. A bracket was made to hold the two cameras and all of the optics rigidly.

2.1.1 Imaging system design

Since crime scene investigators require being able to photograph over a range of working distances and fields of view, we chose to allow imaging of areas from the size of a shoeprint to several square meters. This presented several challenges in the design of the optics for the system. Crucial for the design was the incorporation of a beamsplitter to allow for both cameras to see the same scene and also interference

filters so that each camera only sees the scene at a specific wavelength. Because c-mount camera lenses have a fixed back focal length of 12.552 mm there is not enough space to incorporate the beamsplitter and a filter between the c-mount lens and camera sensor without drastically altering the nature of the focused image. Three different optical configurations were therefore evaluated.

The first design involved having two c-mount lenses directly attached to the cameras with filters in front of the lenses and a beam splitter in the front, as shown in Figure 3.

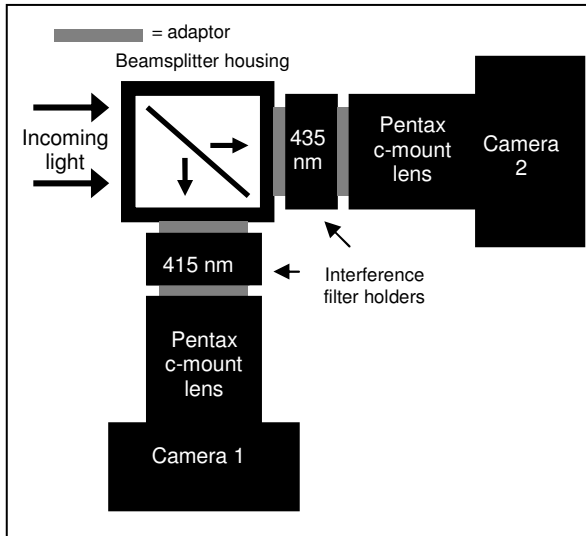


Figure 3: Initial prototype of real-time imager

A second design involved having a single achromat as the imaging lens in front of the beamsplitter. A threaded barrel adjustment placed in front of the beamsplitter allowed for focusing of the image onto the two camera sensors. The path length through the beam splitter, the filter holders, and the threaded barrel was approximately 100 mm. This required the focal length of the imaging lens to be at least this long in order to form an image. A 120 mm achromat lens was used to test the design.

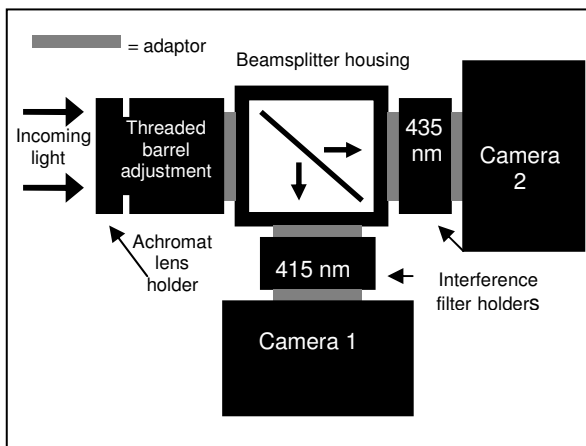


Figure 4: Trial design with single imaging lens in front of beamsplitter.

A third design involved having a c-mount lens mounted to the beamsplitter and two 50 mm achromat pairs (relay lenses) refocusing the image onto the sensors as shown in Figure 5.

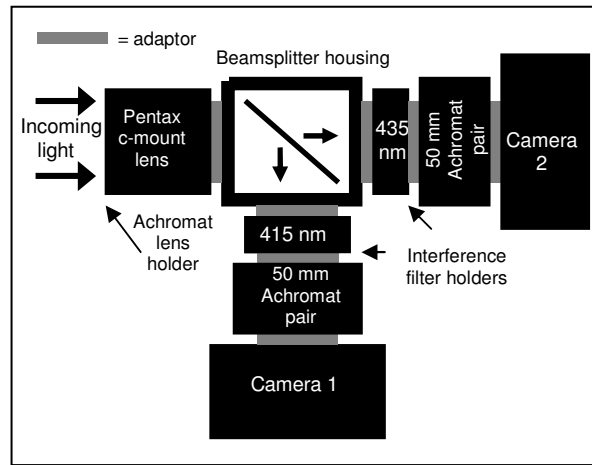


Figure 5: Current design, using single objective with relay optics

The three different configurations were evaluated by imaging a target with several cross hair patterns generated in Photoshop and printed onto a piece of A4 paper. A xenon light source was used to illuminate the target. The resulting individual image quality as well as the final processed image quality of the three configurations was compared to a reference image that was taken by a single Qicam camera with a Pentax lens and 435 nm interference filter on the front.

2.1.2 Lighting

All designs were initially evaluated using a Polilight® (Rofin Australia) as a light source. The Polilight uses a 500 W Xenon bulb which provides a significant amount of light in the blue (as well as the more harmful ultra-violet region). Since the requirements for the project meant that the light source needed to be portable as well as provide the required amount of light at the two wavelengths we then focussed on developing an LED-based lightsource. We were able to locate an LED with a peak wavelength at 435 nm, however finding an LED with a peak wavelength near 415 nm proved to be more difficult and we were forced to try LEDs with a peak wavelength of 420 nm instead. A trial LED flash unit was constructed from six LEDs at a nominal wavelength of 435 nm and six LEDs at 420 nm mounted separately. Both types of LEDs could sustain a current of 100 mA over 30 ms with several seconds allowed for cooling. The LEDs were grouped together and mounted to a breadboard. A circuit was designed to provide the 100 mA pulse to the LEDs.

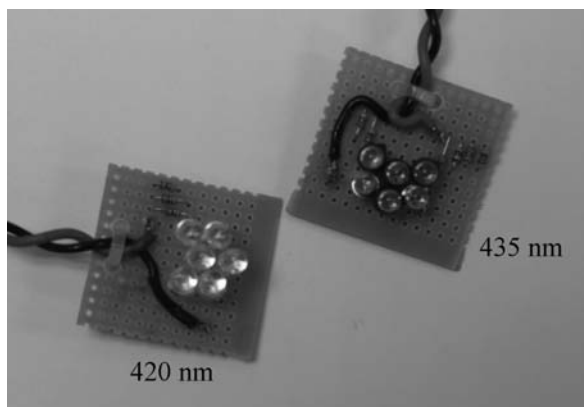


Figure 6 Image of the LED flash unit showing two groups of six LEDs

2.1.3 System integration

For the camera to be a practical handheld system the exposure times should be kept below 30 ms to avoid motion blur. This required careful synchronising between the flash and the two cameras. The two cameras can be made to expose simultaneously either through the control software or through an external hardware trigger. The flash could be synchronised to the cameras either through an external pulse coming from the camera or the flash itself could trigger the cameras to expose through their external trigger option. The circuit providing the power to the flash unit was modified to provide the necessary pulse to the cameras hardware trigger.

To test the synchronisation between the two cameras, a digital stopwatch with a millisecond counter was illuminated with sufficient light and the exposure times of the two cameras were set to 0.5 ms. Using camera configuration 1 the stop watch was imaged while counting up. The degree of synchronisation could be judged by observing the time at which the stopwatch had been imaged. The synchronisation between the LED flash and the camera was judged by setting the cameras to approximately 10 ms, shining the flash against a white piece of paper and imaging the result using camera configuration 1. A bright spot in the image was indicative of successful synchronisation between the flash and the two cameras.

2.1.4 Image processing/software

The image processing required mirroring the reflected image from the beam splitter, an automated series of operations to correct for image mis-registration, and a division of the 415 nm image by the 435 nm image in accordance with our two wavelength method [4]. The final procedure required is a contrast adjustment to improve visualisation. Currently the simultaneous image capture is performed in QCapture from Qimaging Corp. The mirroring, alignment and

division are then all performed in V⁺⁺. We are currently working towards having simultaneous image capture also performed in V⁺⁺. Once the images had been captured into the computer, processing time was approximately two to three seconds which satisfied our criteria for near-real-time imaging.

The automated alignment procedure was achieved by imaging a printed set of cross hair targets and recording the x,y pixel coordinates of the cross hair targets in each image. The 415 nm image was chosen as the reference image and the 435 nm image was aligned to it. V⁺⁺ is able to perform global translation, rotation and scale to achieve image registration. Once the parameters of translation rotation and scale were extracted they could be integrated into the image processing procedure with no input required from the user. To avoid significant truncation when performing the division both images were converted into 32-bit floating point format. Following division the resulting image was multiplied by a factor of 50000 and converted back to 16-bit format for storage and retrieval.

3 Results

3.1.1 Camera configurations

The various camera designs were evaluated on their ability to provide good image quality (i.e, good contrast and minimal distortion relative to the standard image), sufficient light throughput, and flexibility for the end user. The exposure time for the reference configuration (single camera at 435 nm) was 30 ms providing an average exposure level of 250 pixel units across the image with good focus and no visible distortion.

Design 1 provided the best image quality of the three dual camera designs as the c-mount lenses were directly attached to the cameras as in the reference configuration. The exposure time required for the 435 nm camera was about 65 ms for the same exposure level indicating no significant loss of light. (An exposure time of twice the reference would be expected as the beamsplitter halves the light intensity for each camera) However, this design had two main disadvantages. First having the lenses behind the beam splitter housing lead to significant vignetting which reduced the number of useful pixels. Second, having to focus two lenses independently lead to mis-registration between the two images. The mis-registration was not consistent which meant it had to be dealt with on an image by image basis. This extra time consuming step meant the system could not be used for a near real time application.

Design 2 provided better light throughput than the reference configuration requiring only 8 ms to reach

the same exposure level. However the resulting image quality was poorer since the image was not as sharp and there was distortion towards the edges of the image. The focus and distortion was largely corrected by placing an aperture between the lens and the camera sensor but this led to a reduction in light throughput to the extent that 130 ms was required to reach the reference exposure level for a sharp image. The two cameras again showed misregistration. However because the two cameras were fixed relative to each other the misregistration remained a constant and could be corrected for in the image processing software. However the long focal length of the lens, necessitated by the long optical path length, resulted in a very narrow field of view and hence very little flexibility for the end user.

Design 3 provided good image quality, and as with design 2 any misregistration is consistent and can therefore be corrected before the images were processed. Having a standard imaging lens at the front of the system also allowed for the focussing flexibility of the reference configuration. One disadvantage with this system was the decrease in light throughput in the system compared to the reference configuration. Thus the exposure was approximately 3 times longer than the reference to reach the same pixel brightness level. This is possibly due to the strongly diverging rays coming from the back of the c-mount lens hitting the walls of the beamsplitter before being collected by the relay lens.

In camera configuration 1 image registration could be achieved through global translation, rotation, and scaling. These operations were easily implemented and automated in V⁺⁺. However the registration of images from camera configurations 2 and 3 could not be achieved using these operations alone. We were able to correct the mis-registration using the 'polynomial' transformation type in MATLAB[®] Image Processing Toolbox. This transformation is not trivial to reproduce in the scripting language (Vpascal) in V⁺⁺ which means the procedure had to be performed in MATLAB[®] initially and we are working towards implementing the procedure in V⁺⁺.

3.1.2 LED flash and system integration

When testing the degree of synchronisation between the two cameras, the software trigger option was found to have a delay of about 100 ms between the two exposures, while the external hardware trigger option provided submillisecond synchronisation between the two cameras. We also found there to be good synchronisation between the flash and the cameras using the hardware trigger method.

The capability of the current LED flash was evaluated using camera configuration 1 and the 420 nm and 435

nm LEDs. The LEDs with peak wavelength of 435 nm and pulsed at 100 mA for 30 ms provided enough illumination to adequately expose an area approximately 10 by 10 cm. Unfortunately the LEDs with the 420 nm peak output did not have sufficient output at 415 nm to be viable for this project. Although the area successfully illuminated by the 435 nm LEDs is small it should be possible to illuminate areas large enough to be useful for crime scene investigators by scaling up the number of LEDs in the flash unit.

4 Discussion

Design 3 appears to meet many of the criteria: sharpness, little distortion, correctable mis-registration. However, the decreased light throughput means that an improved light source is required. The next development will be to prepare a light source with increased numbers of LEDs centred at 415 nm and 435 nm. These LEDs will need to be mounted in such a way as to diffuse the output to provide smooth and even illumination from the two LED types. Some experimentation is still required to determine how many LEDs at each wavelength will be required to provide sufficient illumination.

Work is also to be completed on fully characterising the nature and degree of the image misregistration and image distortion. It was noticed that some defocusing occurs towards the edges of the images in camera design 3. This defocusing could be reduced significantly by reducing the aperture; however this results in a reduction in light throughput through the system. To maintain light throughput we are attempting to correct the distortion through Fourier filtering methods. Finding a suitable balance between optimising the optics and lighting to capture the best images and correcting for distortion using software procedures is yet to be achieved.

4.1 Conclusions and Future Work

Our work suggests that a proof of concept near-real time imaging camera for blood stain detection is a feasible goal. We would like to be able to use smaller and less expensive monochrome cameras, however given our stringent lighting requirements the quantum efficiency of the camera used needs to be reasonable. There has been a steady increase in the availability, variety, and power output of LEDs operating in the deep blue part of the spectrum during the course of this project. This increases our confidence that an LED flash with significant output in the wavelengths required for imaging blood can be constructed. Complete automation of image capture from two cameras and subsequent image processing is yet to be

achieved, partly due to limitations in our current software implementation.

5 Acknowledgements

The authors would like to acknowledge Dr Mark Andrews from the Department of Electrical Engineering for his advice and for providing much needed equipment. We also acknowledge Dr David Wardle from the Department of Physics Auckland University for his advice in configuring the optics and the University of Auckland Research Fund for supporting the project. John Wagner would like to thank the Tertiary Education Commission, ESR Ltd., and Tasman Screens Ltd. for support funding throughout the project.

References

- [1] D.L Exline, C. Wallace, C. Roux, C. Lennard, M.P. Nelson, and P.J. Treado “Forensic applications of chemical imaging: Latent fingerprint detection using visible absorption and luminescence” *J. Forens Sci.*, 48 (5) pp.1047-1053, 2003.
- [2] G.M. Miskelly and J.H. Wagner “Using spectral information in chemical imaging” *Forens Sci Int.*, 115 (2-3) pp.112-118, 2005.
- [3] M. Stoilovic “Detection of semen and blood stains using Polilight as a light source” *Forens Sci Int.*, 51 (2) pp. 289-296, 1991.
- [4] G.M. Miskelly and J.H. Wagner “Background correction in forensic photography I. Photography of blood under conditions of non-uniform illumination or variable substrate color- Theoretical aspects and proof of concept” *J. Forens Sci.*, 48 (2) pp. 593-603, 2003.
- [5] <http://www.optical-insights.com/>
- [6] L. Liu, O. Yermolaieva, W. Johnson, F. Abboud, and M. Welsh, “Identification and function of thermosensory neurons in *Drosophila* larvae” *Nature Neurosci* 6 (3) pp. 267-273 2003.
- [7] M. Doi, H. Furusawa, F. Nakata, S.Okamura, M. Sekiguchi, K.Shimasaku, N.Takeyama, “UT 15-color dichroic-mirror camera and future prospects” in *SPIE Proc Vol 3355 Optical Astronomical Instrumentation*, pp.646-657 1998

Watermarking on 3D Model

Chia-Yen Chen¹ and Chi-Fa Chen²

¹ Department of Computer Science, The University of Auckland,
Auckland, New Zealand

² Department of Electrical Engineering, I-Shou University ,
Kaohsiung, Taiwan
yen@cs.auckland.ac.nz , cfchen@isu.edu.tw

Abstract

Different methods of embedding watermarks into 3D model are proposed in this paper. The 3D model is transformed into a 2D matrix by first transforming the 3D rectangular coordinates of the model into cylindrical coordinates with constant interval in the Z axis and quantizing radial angles with the required angular change. 2D watermarks are respectively embedded into the resulting 2D representation of 3D model in spatial and frequency domains for comparison. Orthogonal watermarks are also embedding into 3D models. JPEG-like compression and reduction in 3D points are used as attack processes in testing the robustness of the embedded watermarks. Experimental results show that the watermarks are more robust towards attacks from the same domain.

Keywords: 3D model, point cloud, watermark, DCT

1 Introduction

Digital contents are becoming increasingly popular with the progresses in multimedia processing, information and network technologies. The right protection of digital contents becomes an important issue with regards to the protection of intellectual properties. Watermarking is an effective approach to the copyright protection of digital contents. Previous researches have shown a wide variety of methods for embedding watermarks into 2D images [1,2,3,4,5,6], but little have been done on the watermarking of 3D data, especially for 3D point cloud models [7,8,9], which are popular 3D shape representations [10,11].

This paper presents several methods for embedding watermarks into 3D point cloud models. A 3D model is generally represented by a number of points having 3D coordinates in, for example, the Cartesian coordinate system. The coordinates of the original 3D data may sparsely span a wide range of space with no regularities between the points. Hence, the 3D data are not typical digital data in the sense that data points are quantized in values of the coordinates and the amplitudes. Computations of such irregular 3D models may be costly. Therefore, digitization of the data is required to facilitate processing of the 3D data. As a continuation of our previous works, we take the inherent advantage of the constant interval in the depth dimension (z) in the 3D data [12]. The digitized 3D model is obtained by first transforming the model's 3D rectangular coordinates into cylindrical coordinates, followed by quantizing the

radial angles with the required angular resolution. The digitized 3D model can be represented by a 2D matrix with constant intervals in θ and z coordinates, as discussed in Section 2. In Section 3, we describe the details of embedding 2D watermarks into the respective 2D representations of a 3D model in both the spatial and frequency domains. The extraction of 2D watermarks is essentially the inverse of the embedding process. In Section 4, the robustness of the embedded watermarks is tested, using JPEG-like compression and reduction in 3D points as attack processes. The imperceptibility, robustness, and error rates of embedding the watermarks are also shown and discussed.

2 2D representation for the 3D Model

The proposed algorithms watermark a 3D point cloud model in spatial and frequency domains, respectively, for comparison purposes. Since the data in a 3D point cloud model are generally given in 3D Cartesian coordinates. The x and y coordinates are considered continuous in the sense that they are not limited to a finite set of data. A digitization process is therefore used to quantize the coordinates to desired positions, such that the model's representation becomes regular. Regardless of the algorithm used, coordinates of each voxel in the 3D model need to be digitized for embedding and extraction process later on.

The digitization process starts by transforming rectangular coordinates into cylindrical coordinates with

constant z interval. The transformation relationships are shown in Eqns.(1) and (2).

$$r_c = \sqrt{X^2 + Y^2} \quad (1)$$

$$\theta = \tan^{-1}\left(\frac{Y}{X}\right), \quad 0 \leq \theta < 2\pi \quad (2)$$

The θ values are quantized to the desired resolution, $2\pi/m$ in the quantization step, where m is the number of points of the 3D model taken along the θ direction. So that the θ coordinate of the j^{th} point is given by Eqn.(3).

$$\theta_j = \frac{2\pi j}{m}, \quad j = 0, 1, 2, \dots, m-1 \quad (3)$$

The 3D point cloud model is then represented by the matrix A , given in Eqn.(4). Elements in a same row in matrix A have the same z coordinates, while elements in the same column have the same θ coordinates. For example, elements in i^{th} row have the same z coordinate, z_i , and elements in j^{th} column have the same θ coordinate, θ_j . The z and θ intervals between two adjacent points in the digitized 3D model are equal.

$$A = \begin{bmatrix} r_{00} & \cdots & r_{0j} & \cdots & r_{0m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ r_{i0} & \cdots & r_{ij} & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{n0} & \cdots & \cdots & \cdots & r_{nm} \end{bmatrix} \quad (4)$$

3 Watermarking Algorithms

The algorithms for embedding the watermarks are respectively carried out in spatial and frequency domains to compare the effectiveness in different domains. A smaller zone for embedding the watermark in the 3D model is chosen by a subsampling process.

3.1 Spatial Domain Algorithm

In this algorithm, 2D binary images of dimensions 64 pixels by 64 pixels, as shown in Fig. 1, are used as the watermarks. The sizes of the images can be varied as required.

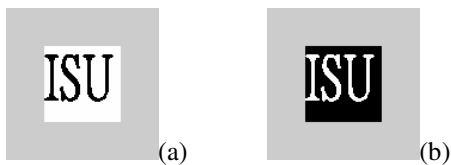


Fig. 1. The 64 by 64 watermarks used in the spatial domain algorithm.

The watermark is added directly to the chosen area in the digitized 3D model representation, i.e. the submatrix B of the matrix A , to produce the watermark embedded matrix A' , where B has the same size as the adopted watermark. Matrix B will be saved as a key for the extraction of the watermark. Matrix A' is then inversely transformed to the 3D point cloud representation in rectangular coordinate system. The 3D Beethoven head model in 3D point cloud format is used for demonstration. Figure 2 shows the side view (a) and bottom view (b) of the original model. Figure 3 shows the side view (a) and bottom view (b) of the watermark embedded model. Translations in some of the data points can be observed by closely comparing the points in Fig. 3 with those in Fig. 2. However, the differences are not significant.

The total error between points in the 3D models before and after embedding is given by the differences in Euclidean distances between corresponding points as shown in Eqn.(5).

$$E = \frac{1}{N} \sum_{i=1}^N \sqrt{(x(i) - x'(i))^2 + (y(i) - y'(i))^2} \quad (5)$$

where E is the measured error,

N is the total number of points in the model,

$x(i), y(i)$ are the coordinates before embedding,

and

(4)

$x'(i), y'(i)$ are the coordinates after embedding.

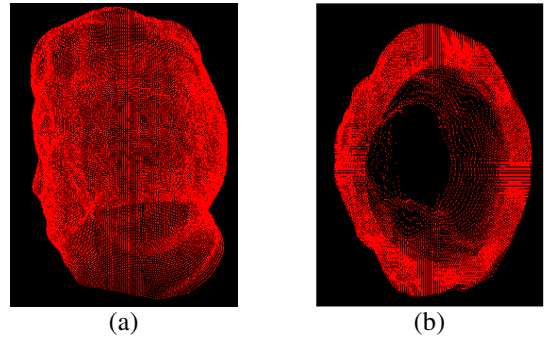


Fig. 2. 3D Beethoven head model.

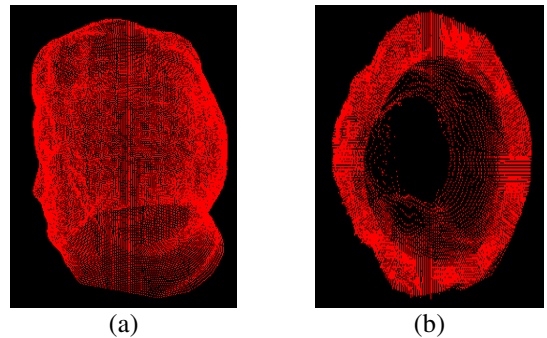


Fig. 3. Watermark embedded 3D Beethoven model (spatial domain algorithm).

The overall error of the overall watermark embedded model shown in Fig.3 has a value of 0.0410. Extraction

of the watermark is readily carried out using matrix B as the key in the reverse procedure.

3.2 Frequency Domain Algorithm

This algorithm uses DCT to convert the point data in matrix A into frequency domain. The watermark used in this experiment has been reduced to 32 pixels by 32 pixels to match the limited number of coefficients in the chosen frequency band of the transformed matrix A .



Fig. 4. The 32 by 32 watermarks used in frequency domain algorithm.

The watermark is embedded in the median frequency band of the transformed coefficient matrix, and indices of the embedded positions are saved for the extraction process. Figure 5 shows an example of the indices selected in the frequency domain. The matrix on the left shows the ordering of the DCT coefficients, and the matrix on the right shows the selected indices.

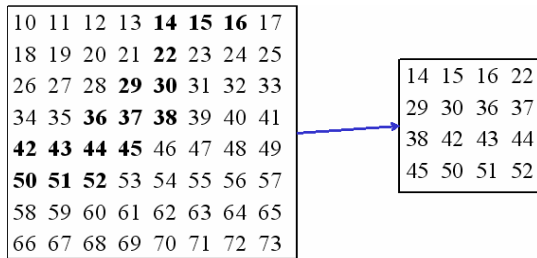


Fig. 5. The indices of the chosen frequency band.

Figure 6 shows the side view (a) and bottom view (b) of the 3D model with watermark embedded using the frequency domain algorithm. Point translations in Fig. 6 are less noticeable than those in Fig. 3. However, the total error between Euclidean distances of the points has a greater value of 0.1213.

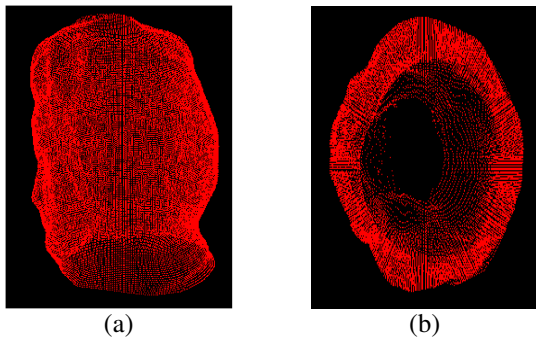


Fig. 6. Watermark embedded 3D Beethoven model (frequency domain algorithm).

4 Attack to Watermarks

In this section, JPEG-like DCT compression and point reduction of the 3D model are used to test the resilience of the watermarks embedded in 3D models. The mean absolute error (MAE) shown in Eqn.(6) is used compare the original and extracted watermarks.

$$MAE = \frac{1}{M * N} \sum_{i=1}^M \sum_{j=1}^N |a(i, j) - b(i, j)| \quad (6)$$

where $a(i, j)$ is the pixel value of the original watermark at coordinate (i, j) , and $b(i, j)$ is the pixel value of the extracted watermark at coordinate (i, j) .

There are two kinds of attacks used to test these two watermarking algorithms, resulting in a total of four different combined cases.

4.1 Results of Point Reduction Attack

A. Spatial domain algorithm

The total number of points in the original 3D model is 41866. The test reduces the number of points and calculates the MAE for each extracted result. Table 1 shows the results for the two watermarks using spatial domain algorithm. The corresponding extracted watermarks are shown in Figs. 7(a)-(d). The watermarks become invisible when the points are reduced to about half as shown in Fig. 7(d).

Point number	MAE
35000	0.0470 (a)
30000	0.0789 (b)
25000	0.1138 (c)
20000	0.1548 (d)

Table 1. MAE of watermark extracted using spatial domain algorithm with point reduction.

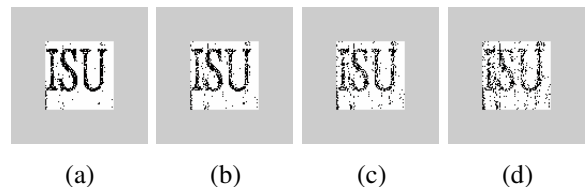


Fig. 7. Watermarks extracted using spatial domain algorithm with point reduction.

B. Frequency domain algorithm

The size of watermark used here is 32 by 32 as shown in Fig. 4. The 32 by 32 entries in the median frequency band of the DCT transformed A matrix are chosen to embed the 32 by 32 watermark. Again, the number of points is reduced and the MAE is calculated for each

extracted result. Table 2 shows the results for the two watermarks using the frequency domain algorithm. The corresponding extracted watermarks are shown in Figs. 8(a)-(c). Similarly, the watermarks become invisible when the number of points is reduced to about half as shown in Fig. 8(c). The extracted watermark shown in Fig. 8(c) appears to be worse than in Fig. 7(c).

Point number	MAE
35000	0.0176 (a)
30000	0.0654 (b)
25000	0.1338 (c)

Table 2. MAE of watermark extracted using frequency domain algorithm with point reduction.

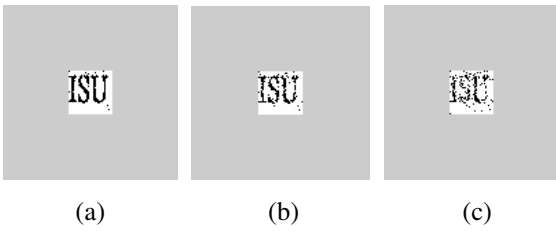


Fig. 8. Watermarks extracted using spatial domain algorithm with point reduction.

4.2 Results of Compression Attack

A. Spatial domain algorithm

Different compression rates are applied to the watermark embedded matrix A . The MAE of watermarks in the decompressed 3D models are calculated. Table 3 shows the results for the spatial domain algorithm. The corresponding extracted watermarks are shown in Figs. 9(a)-(c). The watermarks are still visible even when the compression rate is about 26 as shown in Fig. 9(d).

Compression rate	MAE
14.6	0.0417(a)
17.5	0.0596(b)
21.5	0.0696(c)
26	0.0947(d)

Table 3. MAE of watermark extracted using spatial domain algorithm with compression.

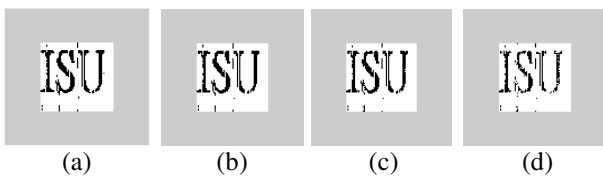


Fig. 9. Watermarks extracted using spatial domain algorithm with compression.

B. Frequency domain algorithm

The different compression rates are applied to the DCT matrix A , with watermarks embedded in frequency domain. The MAE of the extracted watermarks are calculated and shown in Table 4. Corresponding watermarks extracted are shown in Figs. 10(a)-(d). The watermarks are still visible even when the compression rate is about 75 as shown in Fig. 10(d). The result shown in Fig. 10(d) appears to be better than that of Fig. 9(d).

Compression rate	MAE
24.8	0.0586 (a)
40.9	0.0664 (b)
54.6	0.0745 (c)
74.5	0.1055 (d)

Table 4. MAE of watermark extracted using frequency domain algorithm with compression.

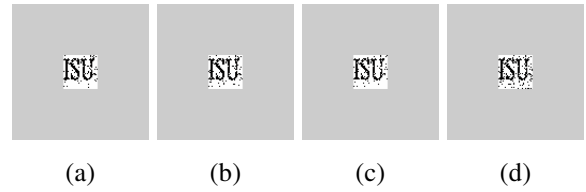


Fig. 10. Watermarks extracted using frequency domain algorithm with compression.

4.3 Results of Orthogonal Watermarks

Two watermarks with orthogonal property as shown in Fig. 1 are applied simultaneously to the 3D model in spatial domain. Results of applying the previous two kinds of attacks are given below.

A. Point reduction Attack

The corresponding watermarks extracted after point reduction attack are shown in Figs. 11(a)-(h). Table 5 shows the results for the two watermarks using spatial domain algorithm.

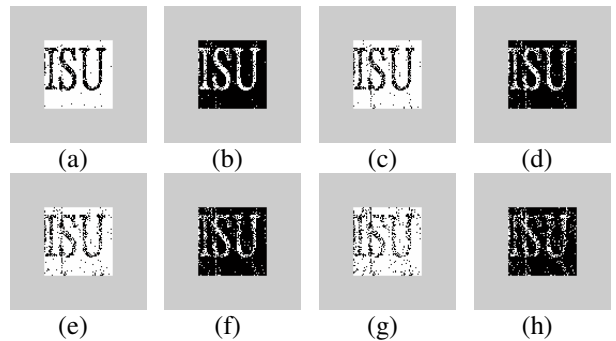


Fig. 11. Watermarks extracted using spatial domain algorithm with point reduction.

Point number	MAE with type (a) watermark	MAE with type (b) watermark
20000	0.1493	0.1516
25000	0.1089	0.1123
30000	0.0707	0.0769
35000	0.0401	0.0481

Table 5. MAE of watermark extracted using spatial domain algorithm with point reduction.

B. Compression Attack

Similarly, watermarks extracted after compression attacks are shown in Figs. 12(a)-(h). Table 6 shows the results for the two watermarks.

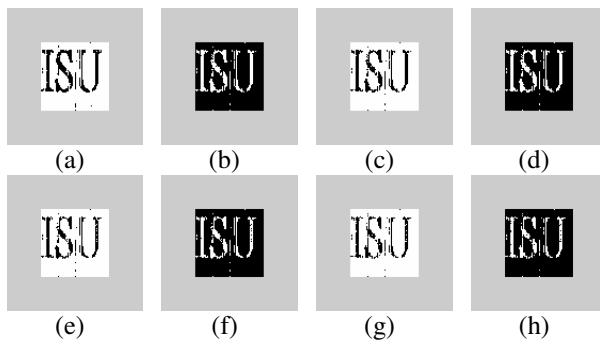


Fig. 12. Watermarks extracted using spatial domain algorithm with compression.

Compression rate	MAE with type (a) watermark	MAE with type (b) watermark
14.6	0.0452	0.056
17.5	0.0545	0.0625
21.5	0.0625	0.0725
26	0.0757	0.0857

Table 6. MAE of watermark extracted using spatial domain algorithm with compression.

5 Conclusion and Discussion

This paper proposes several methods for embedding 2D watermarks into 3D point cloud models. The proposed methods are applied and compared in both the spatial and frequency domains. Reduction of data points and lossy compression are used as attacks to test the resilience of the embedding approaches. Experimental results show that the watermarks embedded in spatial domain have stronger resilience to attack processes carried out in the spatial domain, and the watermarks embedded in frequency domain are more robust towards attack processes in the frequency domain. Further work using 3D watermarks and other watermarks selected subject to the properties of the 3D model will be conducted in the future.

Acknowledgments

This research has been partly funded by the project ISU95-01-02.

References

- [1] M. Bami, F. Bartolini, V. Cappellini, A. Piva and F. Salucco, "The based geometric normalization for robust watermarking of digital maps," International Conference on Image Processing, Vol. 1, pp.1082-1085, 2001.
- [2] C. T. Hsu and J. L. Wu, "Hidden digital watermarks in image," IEEE Transaction on Image Processing, Vol. 8, pp.58-68, Jan. 1999.
- [3] Wen-Nung Lie, Guo-Shiang Lin, Chih-Liang Wu, and Ta-Chun Wang, "Robust Image Watermarking On the DCT Domain," IEEE International Symposium on Circuits and Systems, Geneva, Switzerland, May 28-31, 2000, vol.1, Page(s):228-231.
- [4] H.Kii, J. Onishi and S. Ozawa "The digital watermarking method by using both patchwork and DCT," IEEE Multimedia Computing and System, pp.895-899, Jun 1999.
- [5] C.F Wu and W.S. Hsieh, "Digital watermarking using zero tree of DCT," IEEE Transactions on Consumer Electronics, Vol.46, No.1, pp.8-94, 2000.
- [6] Wen-Nung Lie, Guo-Shiang and Ta-Chun Wang, "DIGITAL WATERMARKING FOR OBJECT-BASED COMPRESSED VIDEO," IEEE International Symposium on Volume 2, 6-9 May 2001, pp.49-52 VOL.2, 2001.
- [7] R. Ohbuchi, A. Mukaiyama, S. Takahashi, "Watermarking a 3D shape model defined as a point set," Proceedings of the 2004 International Conference on Cyberworlds, 18-20 Nov. 2004, Page(s):392 – 399.
- [8] A. G. Bors, "Blind watermarking of 3D shapes using localized constraints," Proceedings of 2nd International Symposium on 3DPVT 2004, 6-9 Sept. 2004, Page(s):242 – 249.
- [9] M. Ashourian, R.Enteshari, J. Jeon, "Digital watermarking of three-dimensional polygonal models in the spherical coordinate system," Proceedings of Computer Graphics International, 2004, Page(s):590 – 593.
- [10] Chia-Yen Chen, Reinhard Klette and Chi-Fa Chen, "Shape from Photometric Stereo and Contours," Proc. of CAIP 2003, August 25-27, The Netherlands, 2003.
- [11] Chia-Yen Chen, Reinhard Klette and Chi-Fa Chen, "3D Reconstruction Using Shape from Photometric Stereo and Contours," Proc. of IVCNZ 2003, November 26-28, New Zealand, 2003 pp. 251-255.
- [12] Chi-Fa Chen and Chia-Yen Chen, "Compression of 3D Point Data Using Discrete Cosine Transform," Proc. of Image and Vision Computing New Zealand 2005, pp.279-284.

License Plate Detection and Classification using a Space Displacement Neural Network

M. Johnson¹, A. Barczak¹, S. Russell²

¹Massey University, Albany, Auckland, New Zealand.

²Knowcam Ltd., Auckland, New Zealand

Email: M.J.Johnson@massey.ac.nz

Abstract

A method is presented for the detection and classification of license plates in real time. The classifier and detector both use a Space Displacement Neural Network which can efficiently be applied to images and is trained using gradient-based learning. The detector has an error rate of less than one percent for individual characters and can find multiple plates in a single image. The classifier has an error rate of less than two percent. The complete system runs at more than 15 frames per second.

Keywords: License Plate Recognition, Convolutional Networks, Space Displacement Neural Network.

1 Introduction

The automatic reading of license plates (Automatic License Plate Recognition or ALPR) is an important task for traffic control and security. There are a number of commercial products available for ALPR, these are mostly based on standard optical character recognition techniques. This paper presents a novel method using a type of neural network for both detection and recognition of the plate.

ALPR involves three main tasks, plate detection, character segmentation and character recognition[1].

Plate detection is the process of finding a plate in an image. This is often done by searching for rectangular regions using standard image processing techniques [2][3].

Character segmentation involves finding a bounding box for each character in the plate. This is commonly done using techniques such as heuristic over segmentation[4].

Character recognition involves classifying the segmented plate region to provide a probability for each possible character class. This is usually done using standard pattern recognition techniques such as Neural Networks or Support Vector Machines.

Figure 1 shows a typical image of a vehicle and plate.

2 Training and Test Sets

In order to train the classifiers, over 4500 images of vehicles were obtained with plates at different angles and scales. The images were obtained over many days to give different types of illumination. From these images, 4800 12x22 pixel images of individual plate characters were hand-segmented and labelled. For plate detection, random background images were added to the set and the characters were shifted plus

or minus one pixel from the centre to give 9 copies of each character and a total of 86,000 images. These images were split into 2 parts, 60,000 for the training set and 26,000 for a test set. The test set contained only plates from images that did not appear in the training set. Figure 2 shows a small part of the test set.



Figure 1: Typical Image

For classification, the character images were shifted up and down by one pixel to give 14,400 images. These were split into an 8000 image training set and a 6400 image test set.

3 Architecture

The architecture is based on a convolutional neural network, this architecture is used by the best performing classifier for hand written characters[5]. In tests on the MNIST database (hand drawn numerals) convolutional networks achieve an error rate of 0.4%, a Support Vector Machine(SVM) achieves 1.4% and a 2 layer Multi-Layer Perceptron(MLP) achieves 1.6%. Convolutional networks have also been used for face and object recognition[6][7] but never for licence plate

recognition. A convolutional network is made from a number of layers which either perform convolution, subsampling or are fully connected. Between each layer is a sigmoid function to provide the non-linearity necessary for training. For the MNIST task, the input image is a 32x32 pixel grey scale image. Six 5x5 convolutions are applied to this image to generate six 28x28 pixel feature maps. These are subsampled to give six 14x14 pixel maps. These maps are then convolved with 62 5x5 convolutions and summed through a partial mapping to give 16 10x10 feature maps. The feature maps are subsampled to give 16 5x5 maps. These maps are fully connected to a set of 120 units which are fully connected to 10 output units, one for each class. For complete details of this architecture see[8].



Figure 2: Part of the Test Set.

A significant advantage of the convolutional approach over other methods is that it allows the network to be applied to large images very efficiently. When a convolutional network is used in this way it is called a Space Displacement Neural Network (SDNN) and is similar to the Time Delay Neural Network (TDNN) used for speech recognition. Figure 3 shows the architecture of a typical SDNN used for classification. The input is an image and the output is a map for each class showing the location of all the pixels in the image assigned to that class. A classifier such as a SVM or MLP needs to be applied to an area around

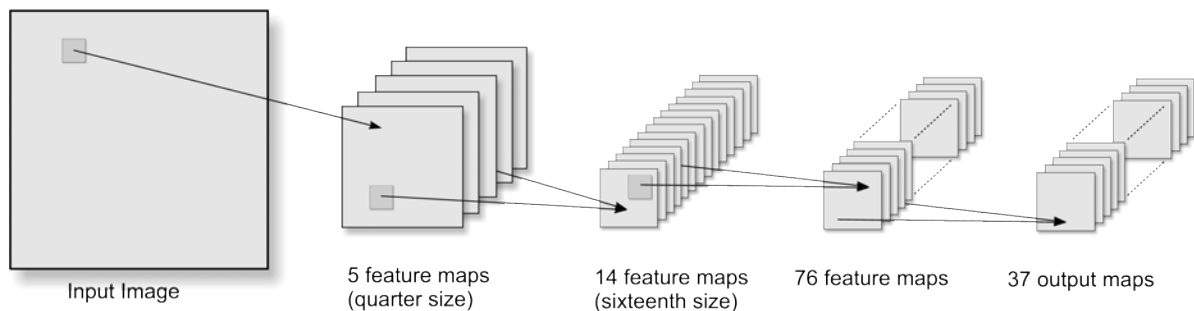


Figure 3: Typical convolutional network architecture for classification.

each pixel in the image separately. This makes these techniques impractical for real time processing of video images without additional hardware support (for example FPGAs).

3.1 Detection architecture

A plate detector needs to be fast and robust, in this case the goal was to achieve over 97% accuracy at over 10 frames per second. For the detector we used a convolutional network with five 1x5 convolutions followed by five 5x1 convolutions followed by a 2x2 subsampling and then five 4x9 convolutions which are summed to give a single output map. The use of one dimensional convolutions means that training effectively finds two separable masks to cover the input region. The one dimensional convolutions can be implemented very efficiently.

The output map was filtered using an integral image to find the sum at every point in the image of a 40x6 area. This is used to remove noise and increase the detection accuracy.

3.2 Classifier Architecture

The classifier needs to discriminate between 36 characters (numbers 0-9 and letters A-Z). For the classifier, we used five 5x5 convolutions followed by 52 5x5 convolutions. This gives 14 maps to which are applied a 2x2 subsampling followed by 1064 2x7 convolutions to create 76 maps. These 76 maps form the input to a fully connected network with 36 outputs, one for each class.

4 Training

Both classifiers were trained using gradient-based learning. The Mean Square Error was used as a cost function and stochastic learning with the diagonal Levenberg-Marquardt method was used for training. These methods have been described extensively elsewhere[8][9]. Training was performed using Lush[10], an interpreted, LISP-like language with support for gradient-based learning. Training of the both the detector and classifier took approximately two hours.

5 Implementation

The detector and classifier were implemented in C++ using Opencv (an open source machine vision library) for some of the image processing functions and for image acquisition. The code is cross platform and was developed using an x86-64 Linux distribution. All the operations were performed using single precision floating point. Profiling of the code shows that over 90% of time is spent performing convolutions. No attempt was made to optimise the convolution code, however, this could be achieved using vector instructions such as SSE or by converting multiple convolutions into matrix multiplications and using optimised BLAS libraries[11]. In order to speed up the system, a first pass was made over the image and a pair of integral images constructed containing the sum and sum of squares for the area above and to the left of each point. From this, the variance in an area can be efficiently calculated. Areas with a small variance will not be part of a plate and these are marked and not included in any of the convolutions. In a practical system, background subtraction could also be used to mark areas as not being part of a vehicle.

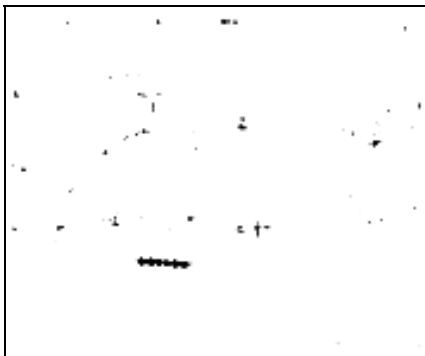


Figure 4: Detection output map for Figure 1.

6 Results

The combined detector and classifier were implemented on a dual core 64 bit Opteron 170 Linux workstation running at 2GHz. The classifier alone ran at 18 frames per second while the combined system ran at 15 frames per second.

6.1 Detection Results

After training of the detector, the training set had an error rate of 0.83% and the test set an error rate of 0.91%. Using a more complex architecture the error rate could be reduced but this would reduce the overall speed of the system. To give a comparison with other methods a Support Vector Machine was trained using the same data and achieved an error rate of 3% on the training and test data.

Figure 4 shows the output map for the image shown in Figure 1. It can be seen that the location of the plate

is clearly indicated and that the six character locations are clearly visible as blobs. There are a number of false detections but these are all isolated and can be easily filtered. Figure 5 shows the filtered output map corresponding to Figure 4. The false detections have been completely removed and the plate centre is clearly visible as a maximum in the filtered output map. When there are multiple plates in an image the output map contains more than one maximum and these can be investigated in turn. Some images, for example a vehicle with lettering on the front or side, give false maximums for these areas. Combining the detector with the classifier should easily be able to rule such areas out as plates. For the prototype system, we used the maximum nearest the bottom of the image as the candidate plate location. Since each vehicle will be seen by the system a number of times and only a single lane of traffic is visible, each plate will appear nearest the bottom of an image at least once. For images covering multiple lanes of traffic it will be necessary to investigate more than one of the maximums.



Figure 5: Filtered output map for Figure 1.

6.2 Classifier Results

After training, the classifier had an error of 0.3% on the training set and 1.86% on the test set. Table 1 shows the results for each character. Some characters such as O, X, I, U and V are not common on New Zealand Plates and the classifier has not seen enough training examples to classify these well. A number of different fonts are used on New Zealand plates and this means that the training set includes some mislabelled patterns. The fact that the training set is classified better than the test set suggests that the classifier is over fitting and the training set is not a large enough sample of plate characters.

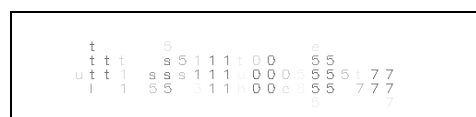


Figure 6: Classifier Output Map

Character	Training set		Test set	
	Errors	%Errors	Errors	%Errors
0	3/291	1.03%	11/234	4.70%
1	0/450	0.00%	5/348	1.43%
2	0/430	0.00%	1/329	0.30%
3	3/480	0.62%	1/363	0.27%
4	0/465	0.00%	1/369	0.27%
5	0/472	0.00%	5/398	1.25%
6	1/407	0.24%	4/334	1.19%
7	0/409	0.00%	3/356	0.84%
8	0/440	0.00%	5/361	1.38%
9	0/406	0.00%	6/347	1.72%
a	0/649	0.00%	2/533	0.37%
b	0/295	0.00%	2/236	0.84%
c	0/319	0.00%	2/251	0.79%
d	0/202	0.00%	2/182	1.09%
e	0/124	0.00%	2/98	2.04%
f	0/118	0.00%	3/116	2.58%
g	0/100	0.00%	2/71	2.81%
h	0/137	0.00%	0/88	0.00%
i	7/16	43.75%	8/11	72.72%
j	0/99	0.00%	3/75	0.00%
k	1/87	1.14%	2/66	3.03%
l	0/95	0.00%	0/76	0.00%
m	3/123	2.43%	3/87	3.44%
n	0/113	0.00%	0/97	0.00%
o	2/56	3.57%	18/43	41.86%
p	0/134	0.00%	0/112	0.00%
q	0/106	0.00%	14/104	13.46%
r	0/115	0.00%	2/95	2.10%
s	0/114	0.00%	0/72	0.00%
t	2/128	1.56%	0/85	0.00%
u	1/96	1.04%	5/78	6.41%
v	1/1	100.00%	2/2	100.00%
w	0/172	0.00%	0/92	0.00%
x	0/50	0.00%	2/43	4.65%
y	0/172	0.00%	1/116	0.86%
z	0/156	0.00%	2/132	1.51%
Total	24/8000	0.30%	119/6400	1.86%

Table 1: Classification Results

The classifier was applied to a rectangular area found by the detector. Figure 6 shows the output map for the plate detected in Figure 1. In order to obtain the actual plate number, liner regression can be used to find the line along which the plate characters are most likely to lie. From this, the Viterbi algorithm can be used to find the most likely character locations and their classifications.

7 Conclusion

Convolutional neural networks are well researched classifiers. They have properties that make them suitable for the efficient detection of objects in images. Using one dimensional convolution masks and a quick test for variance makes this method even faster. This paper has shown that a Space

Displacement Neural Network can be used for practical detection and classification of license plates. Further work is necessary to improve the classification accuracy and to extract the plate number from the classifier output map.

8 Acknowledgements

This work was supported by Knowcam Ltd, Auckland, New Zealand. Training was performed using the Helix parallel computer at Massey University, Auckland.

9 References

- [1] S. Chang, L. Chen, Y. Chung, S. Chen "Automatic License Plate Recognition," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 5, No. 1, pp 42-53 March 2004
- [2] T.D.Duan, D.A.Duc, T.L.Du, "Combining the Hough Transform and Contour Algorithms for detecting Vehicles License-Plates," *ISIMVSP*, pp. 747-750, 2004
- [3] H. Bai, J. Zhu, C. Liu, "A Fast License Plate Extraction Method on Complex Background," *International Conference on Intelligent Transportation Systems*, pp. 985-987, 2003
- [4] T.M.Bruehl, "A System for the off line recognition of hand-written text," *ICPR'94*, pp. 129-134, 1994.
- [5] P. Y. Simard, D. Steinkraus, & J. Platt, "Best Practice for Convolutional Neural Networks Applied to Visual Document Analysis," *ICDAR*, Los Alamitos, 2003, pp. 958-962.
- [6] S. Lawrence, et al. "Face Recognition: A Convolutional Neural Network Approach", *IEEE Transactions on Neural Networks*, vol. 8, No. 1, p. 98-113, 1997.
- [7] O. Matan, C.J.C Burges, Y. LeCun and J. S Denker (1992), "Multi-Digit Recognition Using a Space Displacement Neural Network," *Neural Information Processing Systems, (NIPS)*, 1992.
- [8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.
- [9] Y. LeCun, L. Bottou, G. Orr, and K. Muller, "Efficient BackProp," in *Neural Networks: Tricks of the trade*, (G. Orr and Muller K., eds.), 1998
- [10] L. Bottou, Y. LeCun, "The Lush Reference Manual", <http://lush.sourceforge.net>, 2002
- [11] K. Chellapilla, S. Puri, P. Simard, "High Performance Convolutional Neural Networks for Document Processing", *IWFHR* October 2006

Multiscale Contrast Patterns for Image Tamper Detection

M. K. Bashar¹, N. Ohnishi¹, H. Kudo¹, T. Matsumoto¹, and Y. Takeuchi¹

¹Nagoya University, Dept. of Media Science, Furo-cho, Chikusa-ku, Nagoya, Japan.

Email: khayrul@ohnishi.m.is.nagoya-u.ac.jp

Abstract

With the explosion of digital media, image authenticity becomes a vital issue in our information society. In this research, we propose a statistical model for natural images that is built upon multiscale differential features. Images are decomposed by filtering with the Gaussian derivatives of up to second order. Observations reveal that the micro-patterns in the derivative images may contribute to better texture characterization. We therefore propose a new feature, called Local Contrast Patterns (LCP), which extracts multiple complex patterns through estimating the joint difference distribution of the image derivatives at local regions. A compact statistical description is obtained by stacking the first and second order moments from the multiscale LCP. Fisher linear discriminant classifier is then adopted to discriminate between tampered and authentic natural images. We demonstrated the efficacy of the approach through experiments that showed impressive performance compared to the conventional wavelet statistics with a splicing data set from DVMM, Columbia University.

Keywords: Local contrast patterns, statistical moments, image tampering, classification.

1 Introduction

With the advances in the Information Technology, digital cameras and photo-editing software are becoming ubiquitous. Now it is easy to manipulate digital images and make them difficult to distinguish from the authentic photographs. Images are commonly manipulated by cutting and pasting some regions or objects from one place to another. If the manipulated objects are relatively large in size, simple histogram analysis may provide clues for tamper detection. On the other hand, tampering smaller regions or objects (e.g., human faces, trees, flowers etc.) may pose a great challenge in finding reliable solutions. However, there may be some statistical regularities that distinguish natural images from all possible images. Some examples of statistical models are those based on power spectra [1], [2], [3], Markov random fields [4], [5] or wavelets [6], [7].

Multichannel filtering techniques was proven extremely useful in image compression, image coding, noise removal, and texture analysis. One important reason is that such decomposition preserves statistical regularities that can be exploited. In this paper, we describe a statistical model for natural images that is built upon a multiscale decomposition by Gaussian and its derivative kernels. The model consists of simple statistics (mean, standard deviation) that capture the inherent regularities in the natural images. We then demonstrated how

this model differentiates between natural and unnatural (manipulated) images.

The rest of the paper is organized as follows: In section 2, we introduce the concept of Local Contrast Patterns that are based on multiscale decomposition of images. Section 3 describes the adopted classification principle by Fisher Linear Discriminant (FLD). In section 4, we include the experimental results, while section 5 concludes the paper.

2 Multiscale Decomposition and the Proposed Contrast Patterns

Local image surface at a point p can be approximated by spatial derivatives around that point. The validity of this characterization stems from the Taylor series expansion. If $I(p)$ is an image intensity at $p(=x, y)$, the value at $p + \Delta p$ can be estimated by

$$I(p+\Delta p) = I(p) + \Delta p_x \frac{\partial I(p)}{\partial x} + \Delta p_y \frac{\partial I(p)}{\partial y} + \dots, \quad (1)$$

which states that the derivatives at p can estimate the surface in its neighborhood. Therefore, it can be argued that spatial derivatives characterize the shape of the local surface. They also capture useful statistical information about the image. The first derivatives represent the gradient or "edgeness" of the intensity and the second derivatives can be used to represent bars (or blobs).

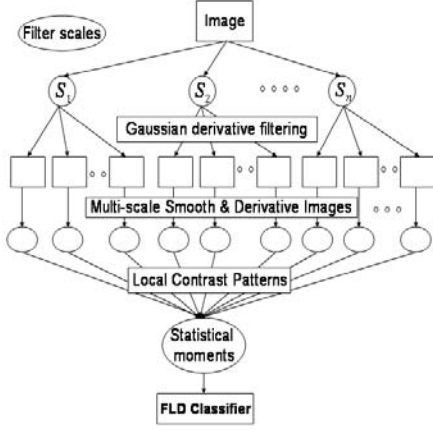


Figure 1: Block diagram of the proposed approach.

Since image tampering operation may alter natural intensity surfaces and its associated statistical properties, we may expect spatial derivatives and/or their associated moments may differentiate between the natural and manipulated surfaces. A simple block diagram of our approach is shown in Fig. 1. We will illustrate our approach in the following subsections.

2.1 Multiscale Decomposition

Multiscale decomposition is essential because one may not know a priori about size, shape, and number of the patterns that exist in the original image. This decomposition can be performed by filtering with the Gaussian and its derivatives at various inner scales as advocated by Koenderink et al. [8]. The linear Gaussian scale-space $L : \Omega \subseteq R^2 \rightarrow R_+ \mapsto R$ of a 2D image $I : \Omega \subseteq R^2 \mapsto R$ is given by:

$$\begin{aligned} L(x, y; \sigma^2) &= \int \int_{\Omega} I(\xi, \eta) G(x - \xi, y - \eta; \sigma^2) d\xi d\eta \\ &= G(x, y; \sigma^2) * I(x, y), \end{aligned} \quad (2)$$

where $L(x, y; 0) = I(x, y)$, σ^2 is a non-negative real number called the scale parameter, $*$ is the convolution operator and $G : R^2 \mapsto R$ is the Gaussian kernel function:

$$G(x, y; \sigma^2) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right). \quad (3)$$

Even though an image, $I(x, y)$ may not be differentiable initially, the corresponding scale-space, $L(x, y; \sigma^2)$, $\sigma^2 > 0$ is infinitely differentiable with respect to (x, y) as long as $I(x, y)$ is bounded. The partial derivative of a scale-space can be obtained by convolving the original image, $I(x, y)$, with the partial derivative of the Gaussian kernel function $G(x, y; \sigma^2)$:

$$\begin{aligned} L_{x^m y^n}(x, y; \sigma^2) &= \partial_{x^m} \partial_{y^n} (G(x, y; \sigma^2) * I(x, y)) \\ &= (\partial_{x^m} \partial_{y^n} G(x, y; \sigma^2)) * I(x, y), \end{aligned} \quad (4)$$

where $\partial_{x^m} \partial_{y^n}$ is a shorthand for $\frac{\partial^{m+n}}{\partial x^m \partial y^n}$. For s^{th} scale, the derivatives can be represented by $L_{x^m y^n}^s(x, y; \frac{\sigma^2}{s})$.

The selection of filter scales is an unsolved problem. It depends on the sizes of the patterns that exist in images. However, natural images often contain many such patterns of variable sizes. In our study, we heuristically choose half octave spacing, that is

$$s = (\sqrt{2})^s \text{ start}, \quad (5)$$

where $s = 0, 1, \dots, s_{max}$. We choose $(0.85, 1.0)$ for start and $(3, 7)$ for s_{max} depending on data sets. The filter kernel-size (FS) is chosen by $FS_s = (\lceil 4s \rceil)^2$.

How many orders of derivatives are suffice? The answer lies with the texture content and pattern complexity of images. However, Ravela [9] pointed out that the first two orders are the most compact features from the information content point of view. We will therefore adopted the first two orders of image derivatives in our study.

2.2 Proposed Contrast Patterns

With the various partial derivatives in hand, the job is now to formulate feature to be extracted. In a recent image retrieval study [10], we observed that the micro-patterns that exist in the derivative responses can characterize texture well. Micro-patterns in an image, regarded as Local Contrast Patterns (LCP), can be estimated by the joint-difference distribution of the derivative(partial) values in local neighborhood regions. For a small neighborhood window N_w (i.e., $b \times b$ pixels), with center at (x, y) , LCP is defined as:

$$\begin{aligned} LCP_{m,n}^s(x, y; \frac{\sigma^2}{s}) &= \sum_{(k,l) \in N_w} w(k, l) \\ &\quad u(L_{x^m y^n}^s(k + x, l + y; \frac{\sigma^2}{s}) \\ &\quad \quad L_{x^m y^n}^s(x, y; \frac{\sigma^2}{s})), \end{aligned} \quad (6)$$

where

$$u(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

Here $L_{x^m y^n}^s(x, y; \frac{\sigma^2}{s})$ is the response image, and $w(k, l)$ is the corresponding weights in the window. For a typical window size (i.e., 3×3), the following binary weights may be used.

$$w(k, l) = \begin{pmatrix} 1 & 2 & 4 \\ 8 & 0 & 16 \\ 32 & 64 & 128 \end{pmatrix}.$$

Note that the LCP value ranges between 0 and 255. For color images, the proposed LCP feature may be computed from Y-component in YIQ transformation [11] by

$$Y = 0.299 R + 0.587 G + 0.114 B, \quad (7)$$

where RGB corresponds to color components in the original image. For the f^{th} derivatives and s^{th} scale, LCP histogram may be defined by

$$h_{lcp}^f(i; s) = \frac{n_i}{N}, \quad (8)$$

where n_i and N are the frequency of i^{th} LCP value and total pixels in $LCP_{m n}^s(x, y; \frac{2}{s})$. The first two statistical moments of contrast patterns are used as texture features for image tampering detection.

$$m_1^f(s) = \sum_{i=0}^{255} i h_{lcp}^f(i; s) \quad (9)$$

$$m_2^f(s) = \sum_{i=0}^{255} (i - m_1(s))^2 h_{lcp}^f(i; s) \quad (10)$$

2.3 Statistical Signature

We constructed statistical signatures by stacking first and second moments from each of partial derivatives of images. Recently, Revela [9] proposed several differential features, (e.g., gradient magnitude (GM), gradient angle (GA), laplacian (LAP), local orientation (LO), isophote curvature (IC), flowline curvature (FC), brightness curvatures (BC), and shape Index(SI)) for image retrieval and recognition. Since these derived features may extract some artifacts of tampering, we will investigate their performance in our study. If we consider all derivatives and their derived features into three groups g_1 , g_2 , and g_3 , the statistical signature can be obtained by:

$$F_{g1} = \left[m_1^{g1}(0), m_2^{g1}(0), \dots, \dots, m_1^{g1}(s_{max}), m_2^{g1}(s_{max}) \right], \quad (11)$$

$$F_{g2} = \left[m_1^{g2}(0), m_2^{g2}(0), \dots, \dots, m_1^{g2}(s_{max}), m_2^{g2}(s_{max}) \right], \quad (12)$$

$$F_{g3} = \left[m_1^{g3}(0), m_2^{g3}(0), \dots, \dots, m_1^{g3}(s_{max}), m_2^{g3}(s_{max}) \right], \quad (13)$$

where $g_1 \in \{L_{x^m y^n}^s(x, y; \frac{2}{s})\}$, $g_2 \in \{LO, BC, SI\}$, and $g_3 \in \{GM, GA, LAP\}$. Note that we

computed first two moments as statistical measures from each feature. These moments may also be computed in local regions without micro-pattern information (MPI). However an empirical study with a data set having texture-texture interface showed poor performance without MPI. Table 1 shows the clear advantages of our approach in terms of the average classification accuracy.

Table 1: Average classification accuracy(training and testing) for the data sets with TT interface.

Features	Average class. acc.(%)			
	Without MPI		With MPI	
	R_{tr}	R_{ts}	R_{tr}	R_{ts}
F_{g1}	77.96	65.23	98.60	66.21
F_{g2}	91.33	58.90	92.62	60.03
F_{g3}	88.95	55.12	89.21	61.92

With micro-patterns, two observations were noted from the table.

- The average classification accuracy becomes larger for almost all features with micro-patterns information.
- The proposed LCP (F_{g1}) attains higher accuracy compared to existing features (F_{g2}, F_{g3}).

Recently, Farid et al. [12] proposed wavelet based higher order statistics for tamper detection. We will therefore use Daubechies D4 based decomposition with first and second order moments, called wavelet statistics(Ws), in our comparative study.

3 Classification

We adopt Fisher Linear Discriminant(FLD) analysis in our study. For simplicity a two-class FLD is described. Denote column vectors \vec{x}_i , $i = 1, \dots, N_x$ and \vec{y}_j , $j = 1, \dots, N_y$ as training sets from each of the two classes. The within cluster means are defined as :

$$\vec{\mu}_x = \frac{1}{N_x} \sum_{i=1}^{N_x} \vec{x}_i, \quad \text{and} \quad \vec{\mu}_y = \frac{1}{N_y} \sum_{j=1}^{N_y} \vec{y}_j. \quad (14)$$

The between-class mean is defined as:

$$\vec{\mu} = \frac{1}{N_x + N_y} \left(\sum_{i=1}^{N_x} \vec{x}_i + \sum_{j=1}^{N_y} \vec{y}_j \right). \quad (15)$$

The within-cluster scatter matrix is defined as:

$$S_w = M_x M_x^T + M_y M_y^T, \quad (16)$$

where, the i^{th} column of matrix M_x contains the zero meaned i^{th} exemplar given by $\vec{x}_i - \vec{\mu}_x$. Similarly, the j^{th} column of matrix M_y contains $\vec{y}_j - \vec{\mu}_y$. The between-class scatter matrix is defined as

$$S_b = N_x(\vec{\mu}_x \ \vec{\mu})(\vec{\mu}_x \ \vec{\mu})^T + N_y(\vec{\mu}_y \ \vec{\mu})(\vec{\mu}_y \ \vec{\mu})^T. \quad (17)$$

Let \vec{e} be the maximal generalized eigenvalue-eigenvector of S_b and S_w (i.e., $S_b\vec{e} = \lambda S_w\vec{e}$). The training exemplars \vec{x}_i and \vec{y}_j are projected onto the one dimensional linear subspace defined by \vec{e} (i.e., $\vec{x}_i^T\vec{e}$ and $\vec{y}_j^T\vec{e}$). This projection simultaneously reduces the within-class scatter while increasing the between-class scatter. Once the FLD projection axis is determined from the training set, a novel exemplar, \vec{z} , from the testing set is classified by first projecting onto the same subspace, $\vec{z}^T\vec{e}$ and then by computing the smallest Euclidean distance between it and the precomputed mean projections of two training sets.

4 Experiments

4.1 Image Data Sets

We collected fairly complex data set from DVMM, Columbia Univ., [13]. It has 933 authentic and 912 spliced image blocks of size 128 × 128 pixels. In our study, we used 336 authentic and 283 spliced images with various object interfaces as shown in Table 2.

Table 2: Data sets for authentic and spliced images.

Name of Authentic Data Sets	Nos.	Name of Spliced Data sets	Nos.
Au-T	126	Sp-T	126
Au-SS-H	50	Sp-SS-H	45
Au-TS-V	100	Sp-TS-V	45
Au-TT-H	60	Sp-TT-H	67
Sub-total	336	Sub-total	283

The image blocks are extracted from images in CalPhotos image set [14]. As the images are contributions from photographers, we consider them as authentic. The authentic category consists of image blocks of an entirely homogeneous textured or smooth region and those having an object boundary separating two textured regions, two smooth regions, or a textured region and a smooth region. The location and orientation of the boundaries are random.

On the other hand, the spliced category contains the same subcategories as the authentic one. The subcategory with entire homogeneous texture is obtained from the corresponding authentic set by copying horizontal or vertical strips of 20 pixels from one location to another within the same images. For the subcategories with object boundaries, image blocks are obtained from images with

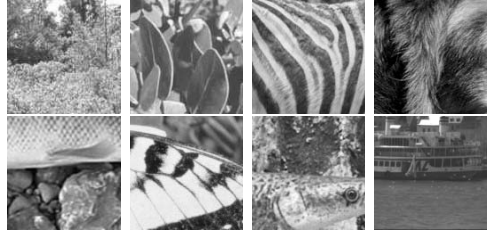


Figure 2: Some samples for authentic images.

spliced objects. Figs. 2 and 3 show some sample images from the authentic and spliced categories. Professional tampered images were made by two steps: (i) splicing, and (ii) post-processing. However, current data sets (except T set) contain many challenging object-interfaces that may be difficult to distinguish by our eyes.

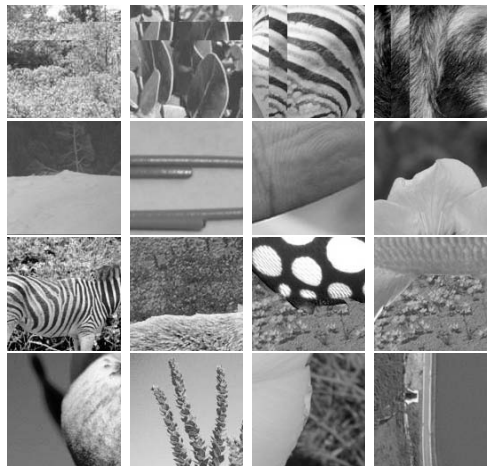


Figure 3: Some samples for splicing images. (row:1) single texture copy-paste tampering, (row:2) SS-H, smooth-smooth interface in hor. dir., (row:3) TT-H, texture-texture interface in hor. dir., and (row:4) TS-V, texture-smooth interface in ver. dir.

4.2 Results

We performed experiments with four spliced data sets: single texture(T), multi-texture sets with smooth-smooth (SS), texture-smooth(TS), and texture-texture(TT) object-interfaces. In each case, classification accuracies corresponding to three different splits of training/testing sets were averaged. Three training sets were fixed sequentially to 25%, 50%, and 75% of the images in each splicing set, while the rests in each case were used as testing sets. We used 8 filter scales for the T data set and 4 scales for other data sets (SS, TS, TT) with 1/2 octave spacing. Tables 3 and figure 4 show classification performance for the proposed and existing features. They confirm that the proposed LCP (F_{g1}) achieves good average accuracy for both training and testing

Table 3: Average Classification accuracy for the authentic and spliced data sets, i.e., SS-H, TS-V, TT-H and T sets.

Features	Classification acc. (%)				Average class. acc. (%)		Object interface type
	Training		Testing		Training	Testing	
	R_{autr}	R_{sptr}	R_{auts}	R_{spst}	$\frac{(R_{autr}+R_{sptr})}{2}$	$\frac{(R_{auts}+R_{spst})}{2}$	
F_{g1}	83.29	84.53	52.43	54.31	83.91	53.37	single tecture with strip copy-pasting
F_{g2}	74.63	70.93	51.04	60.06	72.78	55.55	
F_{g3}	98.24	99.01	38.03	48.16	98.62	43.09	
WS	68.01	64.68	62.84	44.09	66.34	53.46	
F_{g1}	97.36	97.05	65.81	68.14	97.20	66.97	SS interface in horizontal direction
F_{g2}	85.23	79.38	46.58	53.97	82.30	50.27	
F_{g3}	73.39	76.66	55.12	50.21	75.02	52.66	
WS	99.12	98.03	58.11	68.46	98.57	63.28	
F_{g1}	95.55	99.01	68.04	60.84	97.28	64.44	TS interface in vertical direction
F_{g2}	88.00	94.16	71.92	33.63	91.08	52.77	
F_{g3}	91.11	96.07	80.40	50.87	93.59	65.63	
WS	88.88	90.75	74.27	62.34	89.81	68.30	
F_{g1}	98.51	98.69	67.73	64.70	98.60	66.21	TT interface in horizontal direction
F_{g2}	94.07	91.17	67.67	53.59	92.62	60.63	
F_{g3}	88.88	89.54	65.69	58.16	89.21	61.92	
WS	92.59	91.83	56.40	65.03	92.21	60.71	

sets, specifically for data sets with TT interface. Table 4 and figure 5 show the mean accuracy over entire database (T, SS, TS, TT). It shows the following performance order (descending): (i) F_{g1} , (ii) WS , (iii) F_{g3} , (iv) F_{g2} . Below are the observations of our classification experiments:

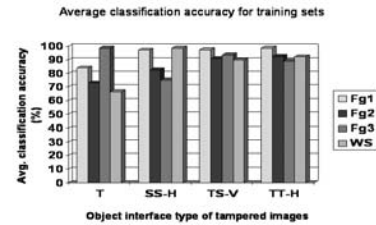
1. Direct derivative based features (F_{g1}) appear better than the derived features from them.
2. Among derived features, edge information statistics (F_{g3}) are more effective than curvature based statistics (F_{g2}).

Table 4: Mean classification accuracy over the entire set for the authentic and spliced images

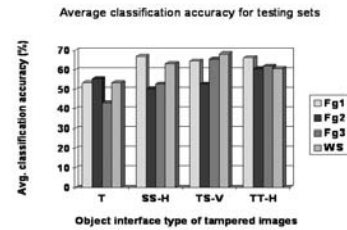
Features	Mean class. acc. (%)	
	Training	Testing
F_{g1}	94.24	62.74
F_{g2}	84.69	54.80
F_{g3}	89.11	55.82
WS	86.73	61.43

4.3 Discussion

In our multiscale approach, the value for the initial scale is selected in trial and error basis, while scale progression is heuristically fixed to 1/2 octave spacing. While it may be possible to obtain better results, we have yet to explore optimization techniques for the effective selection of initial scale and scale progression law.



(a)



(b)

Figure 4: Average classification accuracy for both training and testing sets. Average classification accuracy(%) for the (a) training and (b) testing sets.

We used 3 3 neighborhood in order to extract micro-pattern information. Other neighborhood sizes may also be useful. Optimal feature selection and their combining strategy need to be explored to see if there is any improved outcome.

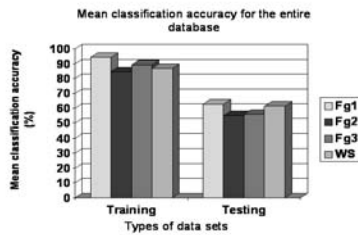


Figure 5: Mean classification accuracy over the entire database.

5 Conclusion

We propose Local Contrast Patterns (LCP) that are based on the joint difference distribution of image derivatives or their derived features. Promising outcomes were obtained by applying the proposed features with Fisher linear discriminant analysis. While it may be possible to obtain improved results, we have yet to explore the scale discretization law with an appropriate value for the initial scale. Selective edge and curvature based features have to be integrated for better classification results. Finally, experiments with a larger database may well justify the feature strength.

Acknowledgements

This work is supported by the Science Research Foundation and the Center of Excellence(COE), Nagoya University, Japan. We would like to thank my laboratory friends for their miscellaneous cooperation.

References

- [1] A. P. Pentland, "Fractal based description of natural scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 661–674, 1984.
- [2] D. L. Ruderman and W. Bialek, "Statistics of natural image: Scaling in the woods," *Phys. Rev. Letters*, vol. 73, no. 6, pp. 814–817, 1994.
- [3] D. J. Field, "Relations between the statistics of natural images and the response properties of cortical cells," *Journal of the Optical Society of America A*, vol. 4, no. 12, pp. 2379–2394, 1987.
- [4] M. Hassner and J. Sklansky, "The use of markov random fields as models of texture," *Computer Graphics and Image Processing*, vol. 12, pp. 357–370, 1980.
- [5] G. Cross and A. K. Jain, "Markov random field texture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, pp. 25–39, 1983.
- [6] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 674–693, 1989.
- [7] E. P. Simoncelli, "Modelling the joint statistics of images in the wavelet domain," in *Proceedings SPIE, 44th annual meeting*, vol. 3814, (Denver, CO, USA), 1999.
- [8] J. J. Koenderink and A. J. van Doorn, "Representation of local geometry in the visual system," *Biological Cybernetics*, vol. 55, pp. 367–375, 1987.
- [9] S. S. Ravela, "On multiscale differential features and their representations for image retrieval and recognition," tech. rep., University of Massachusetts, Amherst, USA, February 2003.
- [10] M. K. Bashar and N. Ohnishi, "Image retrieval by local contrast patterns and color," in *Proceedings International Symposium on Visual Computing (G. B. et al., ed.)*, (Lake Tahoe, Nevada, USA), pp. 1052–1061, Springer-Verlag, 2006.
- [11] L. G. Shapiro and G. C. Stockman, *Computer Vision*. Upper Saddle River, New jersey: Prentice-Hall, Inc., 2001.
- [12] H. Farid, "Detecting hidden messages using higher-order statistical models," in *Proceedings International Conference on Image Processing*, (Rochester, New York, USA), 2002.
- [13] DVMM, Columbia University, "Data set of authentic and spliced image blocks."
- [14] Biodiversity Sciences Technology(BSCIT),University of California, Berkeley , "Calphotos: A database of photos of plants, animals, habitats, and other natural history subjects.." Digital Library Project.

Interactive Styling of Virtual Hair

Rui Zhang and Burkhard C. Wünsche

Department of Computer Science, Private Bag 92019, University of Auckland, New Zealand

Email: burkhard@cs.auckland.ac.nz

Abstract

Interactive styling of virtual hair is an important research field since it is essential for creating realistic looking human avatars for use in virtual worlds, computer games and movie special effects. Virtual hair models can contain thousands of hair strands and hence it is important to develop techniques which enable a designer to efficiently modify the hair in a realistic fashion. In this paper we present a hair styling toolset which uses wisps to represent basic units of hair strands and an improved statistical model for hair wisp generation. The toolset provides a convenient way for users to do operations such as create, edit, delete, copy and paste and hence facilitates the quick creation of hair styles while allowing sufficient control for adding individualistic styling details. The styling process is simplified by using a local coordinate system for hair strands in order to define preferred styling (brushing) directions.

Keywords: hair modelling, interaction techniques, spline curves, key strands

1 Introduction

Computer generated realistic virtual humans are required in applications such as the movie industry (CGI – computer generated imagery), computer games, and as avatars for virtual worlds. An important factor for achieving a realistic human appearance is the development of a realistic hair model. Psychological studies have shown that hair is a determining factor of a person's first impression when meeting his or her counterpart [1]. Therefore, the styling of virtual hair is an active field of research in Computer Graphics. Hair styling is challenging since the complex behaviour of each hair strand and the interactions among the hair strands during animation and styling must be controlled in a physically realistic way.

In order to develop efficient styling tools it is important to discuss different approaches for modelling and rendering hair. Popular approaches to model hair are based on polygonal surfaces, noise-based approaches, volumetric textures, strand-based models, wisp-based models, particle models, and models based on fluid flows or vector fields.

Parke introduced a fast and simple way to model hair which uses simple texture mapped polygonal surfaces to capture the shape and appearance of hair [2]. Many applications nowadays still use this approach due to its simplicity. However, because the surface representation does not model the complex geometry of hair strands the specular lighting effects are not correct and the resulting rendered images lack realism.

Perlin proposed a way to synthesize images of visually complex objects including hair by hypertextures [3]. This noise-based modelling approach cannot capture the movement of individual hair strands. Hence this approach is most suitable for short hair where forces such as gravity and friction between hair strands are small such that the hair hardly moves.

In 1989 a volumetric texture hair model was proposed by Kajiya and Kay [4]. The authors introduced "Texels", which are 3-dimensional arrays of parameters approximating visual properties of a collection of micro surfaces. The most important parameter stored in a Texel is the tangent vector, used to calculate the light reflection by an anisotropic reflection model [4]. Instead of using geometries to model hair strands, Kajiya and Kay use texels to represent hair strands and map them onto the surface of a 3D object. The technique works well for short, furry hair since the corresponding texels are simple and can be generated automatically. However, it is not clear how texels can be easily generated for representing more complex hair styles and how this representation could be used to enable interactive styling.

Strand-based models represent every hair strand explicitly. Because of the large amount of hair strands on a head the strands are most frequently represented by connected line segments rather than connected cylinders in order to reduce the consumption of computing resources when modelling hair [5, 6]. This kind of model is suitable for modelling long animated hair strands with a simple style, but it is not practical for modelling complex hairstyles due to the large

number of strands which must be moved. It is extremely difficult to achieve animation of complex hairstyles because of the complex behaviour and large number of hair strands. Modelling complex hairstyles using a strand-based model is therefore difficult to achieve in real-time.

The key idea of wisp-based models is to group hair strands into wisps and to define their shape and animation using so-called key strands. The idea is based on the observation that adjacent hair strands tend to form wisps due to static attraction and artificial styling products. Daldegan et al. model the underlying head using triangle meshes and use three key hair strands, one at each vertex of a triangle, to interpolate the hair strands of a wisp [7]. Yang et al. use generalized cylinders to represent hair wisps [8]. Plante et al. proposed an animation method to deal with the interactions among wisps and to simulate complex hair motions [9]. The above three models have the advantage that they make it easy to control hair styling. However, the methods are not effective for controlling complex hairstyles such as curly hair. In 2002 Kim and Neumann proposed a multi-resolution hair modelling system, which can handle fairly complex hairstyles [10]. The model makes it possible to define the behaviour of hair over the entire range from hair wisps down to individual hair strands. Different hair styles can be created rapidly using high-level editing tools such as curling, scaling and copy/paste operations. Subsequently Choe and Ko introduced a statistical wisp model to generate a wide range of human hairstyles [11]. The authors simulate hair deformation by applying physical properties of hair such as gravity and collisions detection and response. The model is capable of handling a wide range of human hair styles but is unsuitable for simulating hair animation due to a lack of real-time performance of their modelling algorithm and failing in collision detection in some cases.

Particles, fluid flow, and vector field models for hair were motivated by the observation that slightly curled hair and fluid flows have similar properties in terms of smoothness and continuity. Stam proposed a particle-based hair model, which simulates a hair strand as a trajectory of a particle shot from the head [12]. Hadap and Thalmann considered hair as fluid flow [13], and Yu proposed a hair model using user controllable vector fields [14]. These approaches provide users an easy way to define and modify simple hairstyles, but fail to handle complex ones.

In conclusion we can say that wisp-based models are the most flexible hair models. Additional advantages are their capacity to create a wide range of different hair styles, control details of a hair style, and support high-level operations such as copy/paste between wisps when designing a hair style. Disadvantages of this approach are the large amount of time needed to handle the interactions between hair strands such as collision detection and difficulties in simulating

convincing hair animation. However, for many applications with less animation such as hair styling, these advantages outweigh the disadvantages, and we therefore will use a wisp-based model in our research.

2 A Toolset for Hair Styling

This section first introduces our hair model and then describes the main components of the hair styling toolset and its capabilities.

Our hair model is static and is based on the wisp-based model proposed by Choe and Ko [11]. In order to place our hair strands we use a head model represented by a high resolution triangle mesh. The head model with the scalp region coloured brown is shown in figure 1.

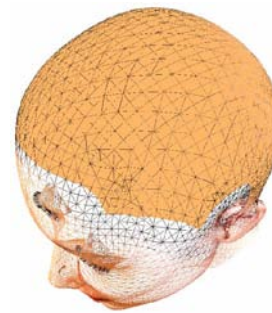


Figure 1: A head model represented by a triangle mesh. The scalp region is coloured brown.

Inspired by Kim and Neumann's interactive hair modelling system [10] we added tools to enable users to manipulate wisps. Hair strands can be grown on the scalp shown in figure 1. Since a scalp can have thousands or tens of thousands of hair strands we need a system to easily control groups of hair strands. The user is able to group several triangles on which to grow a wisp. The size of a wisp (its number of strands) is dependent on the number of selected triangles. The smallest wisp is defined by a single triangle. This level of detail is sufficient for defining a wide variety of hair styles since the triangles are small compared to the scalp's area.

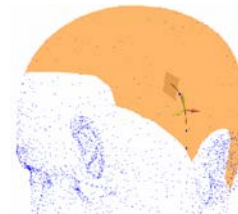


Figure 2: A key strand defining a wisp for a group of four triangles on the scalp. The dark area on the scalp defines the selected triangles. The blue square points are the control points on the key strand.

The geometry of a wisp is controlled using a so-called key strand. A key strand defines the geometry for all strands of a wisp which are then obtained by translating the key strand appropriately. The number of strands for a scalp region depends on the area of

the triangles representing it. We use Catmull-Rom splines [15] to represent hair strands since they are smooth (C^1 continuous) and because they interpolate their control points which makes designing a particular hair style more intuitive. An example of a key strand for a group of four selected triangles of the scalp is shown in figure 2.

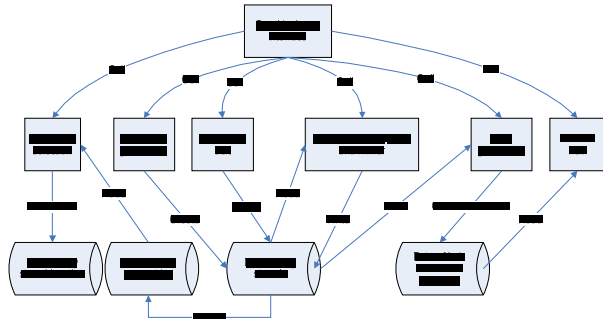


Figure 3: An overview of the components of our hair styling toolset.

The components of our hair styling toolset are illustrated in figure 3. The key strand selection component enables users to choose triangles to grow a wisp or to select an existing wisp for editing. After a group of triangles has been selected it is recorded together with the location of its current key strand. The location of the key strand is determined as the centre of the first triangle of the selected group of triangles (see section 3). The selection of scalp triangles and strand control points has been implemented with the OpenGL “select” mechanism. This enables us to detect whether the projection of a graphical primitive onto the view plane overlaps with a hit region surrounding the mouse location in which case we select the front most primitive.

The key strand generation component enables users to generate one key strand for a selected group of triangles. Users are able to interactively grow a key strand by adding new control points and to modify the 3D shape of a key strand by moving its control points.

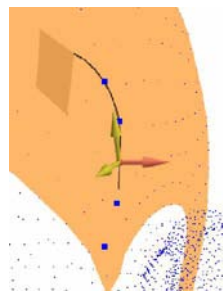


Figure 4: The user interface for changing the 3D positions of a key strand’s control points. The red arrow indicates the currently active direction in which the control point can move forward and backward. The yellow arrows indicate the non-active directions.

Users can change the size of a wisp in the key strand edit component by changing the triangles defining the area on which the strands of this wisp grow. In

addition users can delete an existing wisp and they can change the 3D shape of a wisp by moving, adding or deleting control points of the key hair strand.

The interface for changing the 3D coordinates of the control points of a key strand is illustrated in figure 4. Since the mouse movements on the screen are in 2D we have to map this into a suitable 3D motion. A common solution in modelling applications is to restrict movements to the coordinate directions, parallel to the view plane or within a user defined plane. We found that in hair styling the preferred hair movement direction depends on a particular style, e.g. “brushing” hair backwards, lifting it up, pulling it down or curling it. We therefore define for each key strand a local coordinate system of styling directions. The coordinate system is represented by three orthogonal arrows and the currently active styling direction is indicated by a red arrow. A new styling direction is obtained by choosing one of the non-active arrows or by changing the local coordinate systems as explained below. Suitable default directions for the local coordinate system at a control point are the curve tangent at that point, the surface normal at the scalp point closest to the control point and the vector perpendicular to these two vectors.

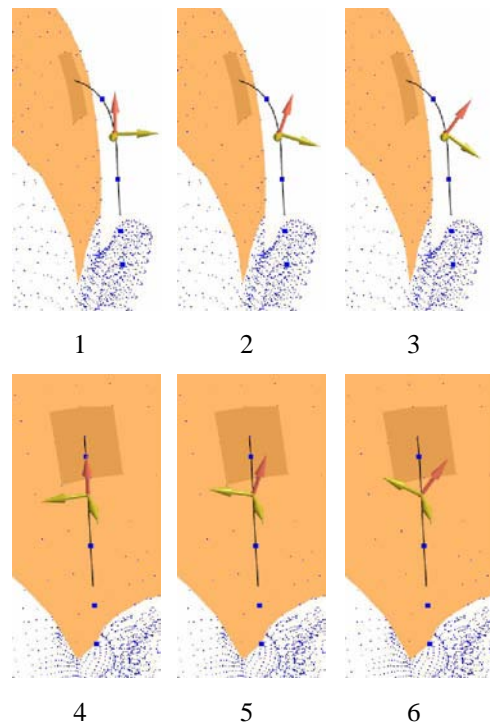


Figure 5: Examples of how the local coordinate system at a control point is changed. The images 1, 2, and 3 show the active arrow being rotated clockwise around the arrow pointing towards the viewer. The images 4, 5, and 6 show the active arrow being rotated clockwise around the bottom yellow arrow.

Since the most suitable styling directions depend on a particular hair style we allow users to adjust the local

coordinate system. Users can modify the local coordinate system by rotating the active arrow around one of the non-active arrows as demonstrated in figure 5.

High-level copy/paste and mirror operations between wisps are provided by the key strand copy/paste and mirror component. After selecting the triangles for a wisp, the geometries of the key strand can be copied or mirrored from an existing wisp's key strand by clicking on it as illustrated in figure 6.

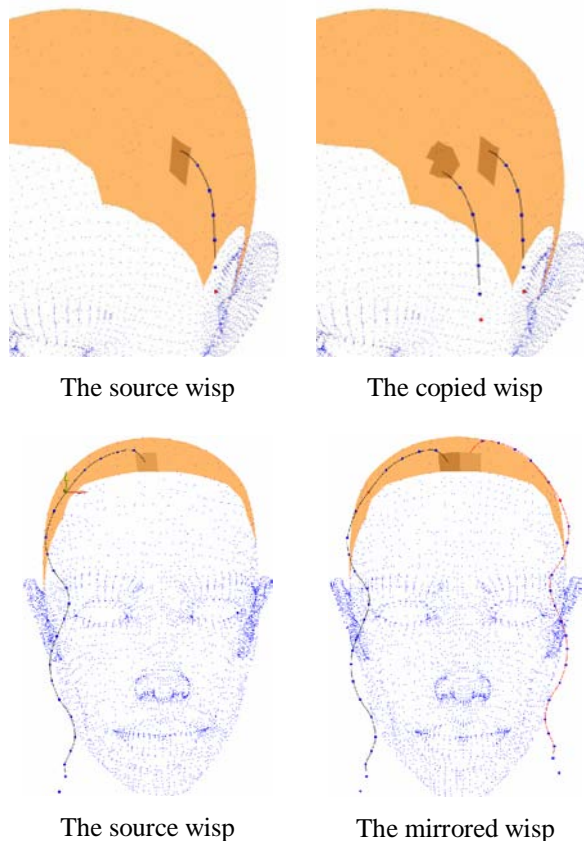


Figure 6: The key strand of the irregularly shaped triangle group in the top right image is an exact copy of the key strand (source wisp) in the top left image. The red key strand in the bottom right image is a mirror version of the key strand in the bottom left image.

The wisp generation component is able to generate all hair strands determined by their key strands and to distribute all hair strands over the scalp uniformly. The geometry of strands is determined using the assumption that the hair strands within one wisp are parallel to each other. The distribution of the hair strands is based on the hair density.

The hair strands within a wisp tend to be similar, although the shapes of the hair strands differ from each other. Choe and Ko observed that the degree of similarity can be controlled by a length distribution, radius distribution and strand variation [11]. The length distribution determines the length variance

between the key strand and a member strand within a wisp. The radius distribution controls the distance between the key strand and a member strand within a wisp. Finally the strand distribution gives the shape variation of each strand compared to the key strand. In our implementation we use a length distribution to control the length of each strand and a novel distance distribution, described in section 3, to control the distance between the key strand and a member strand inside of a wisp. We also implemented a strand distribution but found that the hair styles using it were indistinguishable from the ones using just length and distance variations.

Figure 7 demonstrates how a wisp is generated from a key strand.

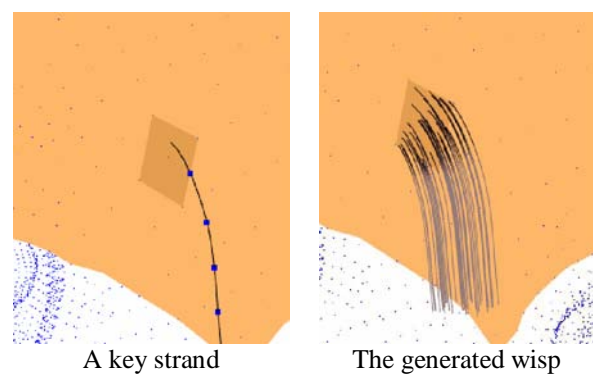


Figure 7: A key strand (left) and the wisp generated from it (right).

The preview hair component gives users a fast overview of the hair style so that users will be able to adjust the wisps of the hair style quickly.

Using our toolset users can create a wide variety of hair styles within a relatively short period of time. Furthermore details of a particular hair style can be changed easily with the toolset. The toolset is designed such that it can be extended effortlessly in future, e.g. by allowing individually coloured wisps simulating coloured streaks of hair.

3 Implementation

The toolset was implemented in C/C++ using the OpenGL library. The most important part of the hair generation component is the creation of hair strands within a wisp according to the key strand.

The shapes of the member strands within a wisp are determined by their key strand. In our implementation we first use the assumption that the hair strands within one wisp are parallel to each other, and then use a length and distance distribution to make individual member strands different from each other. Both of these distributions are based on the Gaussian distribution.

For the length distribution we define the mean as 95% of the length of the key strand which is computed using a simple first order integration method. The variance of the strands' lengths is user defined depending on the desired hair style. We apply a Gaussian distribution to calculate the length of each member hair strand within a wisp but limit the maximum variation to 5% of the key strand. Hence all hairs within the wisp are within 90-100% of the key strand's length and the distribution of the strands' lengths depends on the desired hairstyle (clean-cut look vs. fringy look).

In order to define the distribution of the distances between a key strand and the strands of the corresponding wisp we first define the strands' root positions using a uniform distribution of points over a triangle such that the density of hair is constant over the scalp. We then define for each control point an offset vector which linearly increases in length for each subsequent control point. The initial offset vector is randomly selected using a uniform distribution over a sphere. In order to maintain the overall shape of the strand the offset vector is defined with respect to a torsion minimising reference frame for the spline curve representing the strand [16].

Note that our implementation offers several advantages over Choe and Ko's one [11], who use random offsets for each control point. This can lead to slightly wavy strands even if the original key strand is uniformly curved. Furthermore by defining the maximum length of the initial offset vector we can produce very smooth hair where the strands are virtually parallel and very fuzzy hair where the distance between hair strands increases at the end of a wisp.

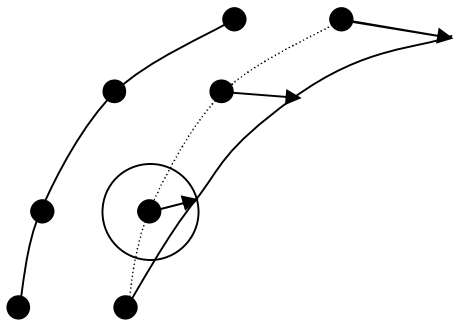


Figure 8: The original key strand (left) and the resulting member strand (right).

Figure 8 illustrates this process. The key strand on the right is reproduced at the new root position. We then define a random offset vector for the first control point subject to a maximum length indicated by the circle in the figure. The offset vector's length linearly increases for subsequent control points. Applying it with respect to the key strand's reference frame generates a new curve of similar appearance.

In order to render hair strands we approximate the Catmull-Rom spline representing them with polylines.

This is achieved by computing curve points at equally spaced parameter values using the Catmull-Rom spline equation

$$P(t) = (0.2 * P_1 + (P_2 - P_0) * t + (2 * P_0 - 5 * P_1 + 4 * P_2 - P_3) * t^2 + (-P_0 + 3 * P_1 - 3 * P_2 + P_3) * t^3) / 2$$

where P is a 3D vertex on the spline segment P_1P_2 , P_0, P_1, P_2, P_3 are the control points for that segment, and $t \in [0, 1]$ is the curve parameter.

4 Results

Our hair styling toolset is capable of creating a variety of moderately complex styles. Depending on the complexity of a new hair style it can take up to several hours for a user without modelling experience to create it. Adjusting the key strands is the most time consuming step when making a specific style. Two examples of completed hair styles created by us are shown in figure 9.



Figure 9: A curly short hair style (left) and a smooth medium length hair style (right) created with our hair styling toolset.

Our hair styling toolset can model real hair styles effectively as demonstrated in figure 10.



Figure 10: Similar hair style generated by the computer (right) and a real human style (left) obtained from [17].

In addition we tested our tool with non-expert users and found that most functions such as wisp copy/paste, mirror, and preview are quite intuitive. However users found that they need to explicitly design the wisp/wisp interactions and it is a little bit difficult to define the directions of key strands. The current version of our toolkit does not perform

collision detection between strands/wisps and does not use an explicit physical model and it is therefore difficult to model braided hairstyles.

5 Conclusion

Although the hair styling process can require a couple of hours we found that our toolset enables users to create a variety of hair styles efficiently and effectively. The toolset provides not only high-level functionality such as copy/paste and mirroring of wisps, but also low-level modifications such as changing the number and positions of a key strand's control points in order to modify the shape of a wisp. This was achieved using a novel interaction tool which uses a local-coordinate system for defining "styling directions".

We have introduced a new statistical method to generate strands from a key strand which has the advantage that it maintains consistency of style within a wisp and that it enables users to model smooth, fuzzy and fringy hair. With our density based hair distribution facility the roots of hair strands are distributed evenly over the scalp.

Rendering is performed in real-time using GPU accelerated algorithms and the whole modelling process is interactive.

References

- [1] M. Lafrance, "First Impressions and Hair Impressions", Unpublished manuscript, Department of Psychology, Yale University, New Haven, Connecticut. http://www.physique.com/sn/sn_yale-study2.asp, visited on 15th July 2005.
- [2] F. I. Parke, "A Parametric Model for Human Faces", PhD thesis, University of Utah, Salt Lake City, UT, UTEC-CSc-75-047, December 1974.
- [3] K. Perlin, "Hypertexture", *SIGGRAPH Proceedings*, pp 253-262, 1989.
- [4] J.T. Kajiya, T.L. Kay, "Rendering Fur with Three Dimensional Textures", *SIGGRAPH Proceedings*, vol. 23, pp. 271-280, July 1989.
- [5] K. Anjyo, Y. Usami, and T. Kurihara, "A simple method for extracting the natural beauty of hair", *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, pp. 111-120, 1992.
- [6] R. Rosenblum, W. Carlson, and E. Tripp III, "Simulating the structure and dynamics of human hair: Modelling, rendering and animation", *The Journal of Visualization and Computer Animation*, vol. 2, no. 4, pp. 141-148, October-December 1991.
- [7] A. Daldegan, N. M. Thalmann, T. Kurihara, and D. Thalmann, "An integrated system for modelling, animating, and rendering hair", *Eurographics Proceedings*, vol. 12, pp. 211-221, 1993.
- [8] X. D. Yang, Z. Xu, J. Yang, and T. Wang, "The Cluster Hair Model", *Graphical Models*, vol. 62, pp. 85-103, 2000.
- [9] E. Plante, M. P. Cani, P. Poulin, and K. Perlin, "A layered wisp model for simulating interactions inside long hair", *Proceedings of Eurographics Computer Animation and Simulation 2001*, pp. 139-148, Sep 2001.
- [10] T. Kim, U. Neumann, "Interactive multiresolution hair modelling and editing", *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, ISBN 1-58113-521-1, ACM Press, New York, NY, USA, pp. 620-629, July 2002.
- [11] B. Choe, H. Ko, "A statistical Wisp Model and Pseudophysical Approaches for Interactive Hairstyle Generation", *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 2, pp. 160-170, Mar-Apr 2005.
- [12] J. Stam, "Multi-Scale Stochastic Modelling of Complex Natural Phenomena", PhD Thesis, Department of Computer Science, University of Toronto, 1995.
- [13] S. Hadap, N. M. Thalmann, "Interactive hairstyler based on fluid flow", *In Proceedings of Computer Animation and Simulation*, pp. 87-100, 2000.
- [14] Y. Yu, "Modelling realistic virtual hairstyles", *Pacific graphics*, pp. 295-304, 2001.
- [15] E. Catmull, R. Rom, "A Class of Local Interpolating Splines", *Computer Aided Geometric Design*, R. E. Barnhill and R.F. Riesenfeld ed., Academic Press, New York, 1974, pp. 317-326.
- [16] J. Bloomenthal, "Calculation of reference frames along a space curve", *Graphics Gems Vol. 1*, Academic Press, San Diego, CA, USA, 1990, pp. 567-571.
- [17] HairBoutique.com – Style #3787, <http://gallery.hairboutique.com/details.asp?ID=3787>, visited on 22nd September 2006.

Classification of 3D LIDAR Point Clouds for Urban Modelling

E. H. Lim and D. Suter

Institute for Vision Systems Engineering,
Monash University, Victoria, Australia

Email: eehui.lim@eng.monash.edu.au

Abstract

Recently, urban modelling with LIDAR (Light Detection and Ranging) data has received much attention and much progress has been made in this area. However, building modelling with terrestrial laser scanned data is still a difficult problem in environments containing multiple structures and vegetation. In this paper, we focus on classifying 3D LIDAR data into man-made structures, terrain and vegetation. The proposed algorithm combines the extraction of distinct features from the point clouds with a discriminative graphical probabilistic model (Conditional Random Field) for classification. We validated our method with urban data acquired from a terrestrial Riegl laser scanner and the result showed that it performs better than a Bayesian classifier.

Keywords: Classification, point clouds, LIDAR, terrestrial, vegetation removal, Conditional Random Field

1 Introduction

Accurate 3D urban modelling from LIDAR (Light Detection and Ranging) data is in demand with the growing number of applications such as regional planning, virtual reality, precise navigation and disaster management. With the great amount of time and work required to reconstruct large scale urban models manually, automatic surface modelling from the urban data is an important research area.

Common methods employed in automatic surface reconstruction directly use triangle meshes[1]. However, direct triangulation carries no information on the modelled structures. Moreover, occlusions, noise, varying densities, multiple structures and the level of complexity in the data acquired from the real world environment make urban modelling a difficult and challenging problem.

In addition, structures that exist in urban data include solid objects, such as terrain and buildings, and porous objects, such as vegetation, that require different processing approaches. Solid objects need to be represented with polygonal meshes with no holes and data spikes (or better still, geometric primitives such as boxes and cylinders), whereas vegetation can be represented by point clouds (with data reduction, possibly at reduced resolution) or may be replaced with generic models. Therefore a useful first stage is to separate vegetation and terrain from the building structures.

Automatic segmentation of vegetation is not new in the literature on urban model data classification. However, most approaches are only useful for airborne LIDAR (Light Detecting and Ranging) data which are 2D (or 2.5D) where filtering via changes in the height difference is possible.

With a Riegl LMS-Z420i terrestrial laser scanner, we need a classification technique that deals with 3D data. In this paper, we focused on discrimination between vegetation, terrain and man-made structure with the terrestrial LIDAR data. The classification result from our framework shows an accuracy of 80% to 90% depending on different data sets.

2 Previous Work

2.1 Covariance as a Region Descriptor

The first step in data classification is to extract features that will hopefully capture the relevant relationships among observations and to label training sequences for the learning model. One of the popular features exploited in recent work is the estimated covariance along the least dominant principal direction of a number of neighbouring data, for identifying locally planar points.

For instance, in [2], Stamos and Allen determined the planarity of each point by thresholding the eigenvalue corresponding to the covariance matrix of k neighbouring data for each point. However, this approach is not applicable to our data containing multiple structures with different sampling rates.

Instead of a fixed window size, Unnikrishnan and Hebert [3] computed the minimum eigenvalue of the covariance of voxel, which is a cube containing point clouds with size calculated with AMISE optimal bandwidth. This solves the problem of fixed scale, however as both approaches classify data with a selected threshold for the value of minimum eigenvalue, the threshold has to be trained for different data sets. In addition, the data in one voxel are all classified into one class, therefore in the case where a voxel contains more than one class of data, some of the data, if not all, will be misclassified.

Instead of using the AMISE optimal bandwidth, Lalonde et al. [4] implemented an adaptive scalable neighbourhood size for the calculation of the covariance. Using 3D scale theory developed by Mitra et al. in [5], the method is capable of shrinking the size of k neighbours at high curvature data points, and expanding the k value in planar regions.

Lalonde et al. used all three eigenvalues for the purpose of classifications and derived a saliency feature vector using the relationship between the eigenvalues (details in Section 3).

2.2 Other Local Feature Descriptors

Other than using covariance as a local descriptor, features such as: intensity [6], height [6-9], surface curvature [9], spin image [7], normals [8] and colour [10] are often combined together or treated independently as feature descriptors. The height of the data and the surface normal vectors are useful to discriminate data with similar geometry structure (such as terrain and wall).

The surface normal vector which can be estimated from the raw laser data is a good representation of texture. Hoppe et al [11] proposed estimating a normal at each point by computing the normal to the least square plane fitted to the k nearest points. Similar to the estimation of covariance, in order to estimate a normal (in spite of varying point density, multi-structure and occluded input data), an adaptive support region size is required. The approach applied in [4] can be used in the estimation of surface normals.

Triangulation is a common method employed for the purpose of surface normal estimation. However, direct triangulation is not suitable for noisy outdoor laser data. To overcome this limitation, Dey and Goswami proposed the Big Delaunay triangulation method that uses Delaunay balls which remain relatively big [12], and therefore are capable of estimating accurate surface normals in noisy data. Comparison of its performance with the numerical plane fitting methods for surface normal estimations can be found in [13].

2.3 Learning Model

Given the extracted features, supervised-learning models can be trained to recognize which data type the point clouds belonged to. For instance, instead of using a manually fixed threshold [2, 3], Lalonde et al [4] learned the distribution of the saliency feature with Bayesian classification by fitting a Gaussian Mixture Model using the Expectation Maximization algorithm on hand labeled data.

In contrast with [4] which classified point clouds in real time, we start to process data after all data acquisition in one area has completed. As a result we have the advantage of having the relationship of complete neighbouring data for each point. Instead of locally classifying each point, a more appropriate approach would be using both global and local information, as spatial relationships exists among the input data. Moreover, local classification often leads to isolated false positives and missing false negatives.

A generative model or a discriminative model can be used for sequential classification problems. Popular generative models include: Bayes classifier, Hidden Markov Models and Maximum Entropy Markov Models. These models define a joint probability distribution of the observation and labelling sequences $p(X,Y)$. Another popular approach includes discriminative models such as CRFs[14] and Markov Random Fields which specify the probability of a label given an observation sequence $p(Y|X)$. By modelling the conditional probability distribution instead of the joint probability distribution, the discriminative models do not need to enumerate all possible observation sequences which may not be feasible [14].

Using both generative and discriminative models, Wolf et al. [15] classified 3D points into navigable and non-navigable regions with Hidden Markov models locally followed by global segmentation with Markov Random Fields. Only concrete walkways are classified as navigable regions and others (including grass) as non-navigable regions. The feature used is the difference in the altitude of the point compared to the altitude of its neighbouring points.

With more complicated object classes, Anguelov et al. [7] segmented 3D scan data into four features - ground, tree, building and shrubbery with an associative Markov network (AMN) that allows effective inference using graph-cuts [16]. The 'plane' features include the first two principal components of the 100 points in a cube of radius 0.5 meter, followed by computing the percentage of points lying in the various sub-cubes which are partitioned from the original cube that represent the plane feature. The 'tree' feature is based on a column around each point by computing the percentage of points that lie in a cylinder of radius 0.25 meters. As for 'shrubby', Anguelov use an indicator feature of the height

threshold at 2m from the floor. The experimental evaluation shows that AMN predicted 93% correctly where as an SVM predicted 68% correctly.

Using AMNs, Triebel et al. [17] classified point clouds into window, wall and gutter. The features employed include: the cosine of angles between the local normal vectors, distribution of neighbours and the normalised height of the points. The results confirmed that the AMN outperforms a generative model - Bayes classifier.

3 Our Approach

We believe an accurate and robust classifier should include the extraction of distinct features and the selection of an effective training model for accurate classification of the acquired urban data. We employed conditional random fields (CRF) [14] to discriminate between planar object and cluttered object.

The feature vectors in our approach include the logarithm of the Lalonde’s saliency features [4], the normal vector and the normalised height. The classification works by first extracting vegetation (based solely on logarithm of saliencies) then discriminates between building and terrain (as shown in figure 1).

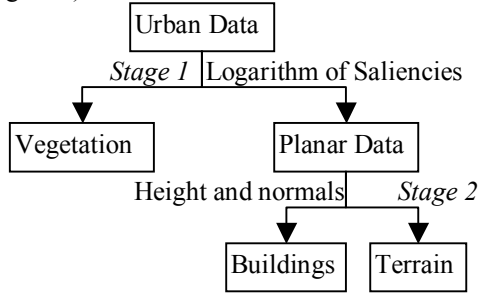


Figure 1: Extracted features for classification

3.1 Saliency Features

Similar to [4], we compute saliency features that capture the spatial distribution of points in a local region (which consist of the k nearest points) to differentiate between planar (man-made objects and terrain) and cluttered data (tree and shrubbery).

The saliency features are derived from the eigenvalues of the covariance matrix of k nearest neighbours for each data point, where the size of k can be varied to address the effect of sampling density difference.

Let $\lambda_1 > \lambda_2 > \lambda_3$ be the eigenvalues of the covariance matrix of the k nearest neighbours. In case of clutter, $\lambda_1 \approx \lambda_2 \approx \lambda_3$ and there is no dominant direction. For points on surfaces $\lambda_1, \lambda_2 \gg \lambda_3$ and for linear structures $\lambda_1 \gg \lambda_2, \lambda_3$. With the relationship between the eigenvalues, the saliency feature can be defined as a linear combination of eigenvalues in the 3-vector [4]:

$$\begin{bmatrix} \text{point-ness} \\ \text{surface-ness} \\ \text{curve-ness} \end{bmatrix} = \begin{bmatrix} \lambda_1 \\ \lambda_2 - \lambda_1 \\ \lambda_3 - \lambda_2 \end{bmatrix} \quad (1)$$

We experimented with the saliency features (as shown in figure 2) of the data acquired, and we found that by taking the logarithm of the saliency features (as shown in figure 3), the classification result is improved (to account for the eigenvalue being sometimes very small).

In order to determine the size of the local neighbourhood, we implemented the iterative procedure explained in [4] and [5] to compute the scalable neighbourhood size. The number of k nearest neighbours depends on the curvature, density and noise of the data points.

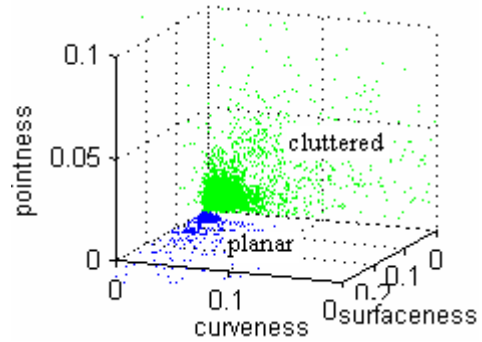


Figure 2: Saliency features of cluttered data (vegetation) and planar data (man-made structure and terrain)

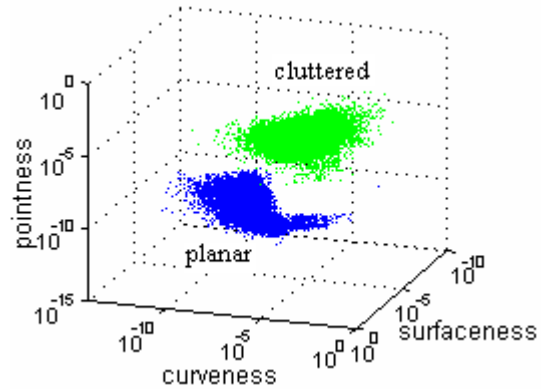


Figure 3: Logarithm of saliency features of cluttered data (vegetation) and planar data (man-made structure and terrain)

The classified planar data points are further segmented into man-made structure data and terrain data (figure 1). The saliency features explained before is not very useful in differentiate these two classes due to the geometry of the both classes are very similar. The most distinct features found in our experiment are the normalised height and the angle between the normal vector and the vertical vector.

The normal vector is obtained by computing the normal to the least square fitted plane of the k nearest

points with the support region as calculated for the saliency features. For a point p , the fitting plane $n^T x = c$ is obtained by minimizing the error term

$$e(n, c) = \sum_{i=1}^k (n^T p_i - c)^2 \text{ with the constraint}$$

$n^T n = 1$ [11]. The normals computed by fitting planes are unoriented. However the distinct difference between terrain and buildings are such that the normals of terrain are more vertical and those of buildings are more horizontal, with the assumptions that the terrain is mostly flat, and the terrestrial laser scanner is unlikely to capture the rooftop of the buildings. Therefore the angle θ between the normal vector and the vertical vector is used as the observation feature instead of the normal vectors.

$$\theta = \arccos(n \bullet [0 \ 1 \ 0]) \quad (2)$$

3.2 Conditional Random Field

CRFs [14] is an undirected graphical model with which promising results have been shown in text processing [14, 18], image segmentation [19, 20], DNA sequence prediction [21], table and diagram structure extraction from documents [22, 23]. We implemented a 1D CRF as the training model and the classification at the sparse density areas are improved.

A special case of CRF is a linear-chain: Let $x=x_1, \dots, x_T$ be the sequence of the observed logarithm of saliency features (or height and angle θ) where the 3D data is raster scanned. Let Y be a set of states, each corresponding to a label $l \in L$ (for example, planar and cluttered; terrain and building). Let $y=y_1, \dots, y_T$ be the sequence of labels in Y given the observable input sequence. The linear-chain CRF with parameters $\Lambda = \{\lambda, \dots\}$ defines the conditional probability for a state sequence given an observable sequence to be:

$$P_\Lambda(y|x) = \frac{1}{Z_x} \exp\left(\sum_{i=1}^T \sum_k \lambda_k f_k(y_{i-1}, y_i, x, i)\right) \quad (3)$$

where Z_x is the normalization constant over all state sequences, that makes the probability of all state sequences sum to one; $f_k(y_{i-1}, y_i, x, i)$ is a feature function which is often binary valued for categorical classes (such as in text applications), but in our application with ordinal observations, the feature function is real-valued; λ_k is a learned weight associated with feature f_k . The feature function is defined over all the local data points feature (for example, the logarithm of saliency features) observation sequence x , the current state y_i and the preceding (spatially) state y_{i-1} .

CRFs learn by finding the weight vector $\Lambda = \{\lambda, \dots\}$ to maximize the log-likelihood. With a Gaussian prior

with variance σ_k^2 , the log-likelihood is penalized as follows:

$$L_\Lambda = \sum_{j=1}^N \log P_\Lambda(y_j | x_j) - \sum_k \frac{\lambda_k^2}{2\sigma_k^2} \quad (4)$$

where the second summation provides smoothing to avoid over-fitting [24]. The scaled conjugate gradient optimization algorithm is used for the maximization.

Given the observation sequence x , inference in CRF is to find a state sequence y which is the most likely

$$y_{\max} = \arg \max_y p_\Lambda(y|x) \\ = \arg \max_y \left\{ \exp\left(\sum_{i=1}^T \sum_k \lambda_k f_k(y_{i-1}, y_i, x, i)\right) \right\} \quad (5)$$

Since exact inference can be intractable in such models, approximate inference using belief propagation is performed for finding y^* .

We manually labelled around two million points of training saliency features for the stage 1 classification and fifty thousand training height and angle θ features for the stage 2 (see section 3.1). Although the urban area are mostly constructed on flat surfaces, the terrain which includes grass and concrete pathway can be sloping, or the orientation of the laser scanner may not be normal to the sea level. This means that setting a simple angle and height threshold for the local classification of terrain and building is insufficient. Therefore we trained a CRF for the segmentation of planar data into terrain and man-made objects (stage 2) that is similar to classification in stage 1. The features in the observation sequence are the angle between the normal vector and the vertical vector, and the normalised height.

4 Results

The result from our approach is compared with segmentation through a Bayesian classifier by training a GMM using the EM algorithm [4] with the logarithm of saliency features as input observation.

The likelihood of a new point x belonging to class C^g is given by:

$$P(f(x) | C^g) = \sum_{i=1 \dots n_g} \left(\frac{w_i^g}{(2\pi)^{3/2} |\Sigma_i^g|^{1/2}} \right) \\ \times \exp\left(-\frac{1}{2}(f(x) - \mu_i^g)^T \Sigma_i^{g-1} (f(x) - \mu_i^g)\right) \quad (6)$$

where

n_g is the number of components of the Gaussian mixture in the g -th class,

$$f(x) = \{\log(\lambda_1); \log(\lambda_2 - \lambda_1); \log(\lambda_3 - \lambda_1)\},$$

$$C^g = \{(w_i^g = \text{weights}, \mu_i^g = \text{means}, \Sigma_i^g = \text{covariances})_{i=1 \dots n_g}\}$$

The predicted class is obtained with:

$$C^{est} = \arg \max_g (P(f(x) | C^g)P(C^g)) \quad (7)$$

The result is generated with the optimal number of Gaussians n_g needed to fit the saliency feature distribution, where $n_g = 3$ (two for planar data and one for cluttered).

The selected results are shown in figure 4 to 9. We can see that the predictions of the CRF (figure 6 to 9) are much smoother: for example the trees near the building and the terrain with sparse density are predicted correctly. We attribute the misclassification at the wall corners being the edge effect in the chosen feature [4] and we hope to improve this with an edge filter.

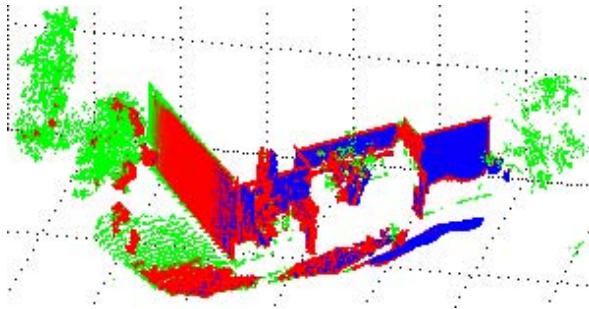


Figure 4: Sample result 1 with GMM (three Gaussians: red and blue for planar data class and green for cluttered data class)

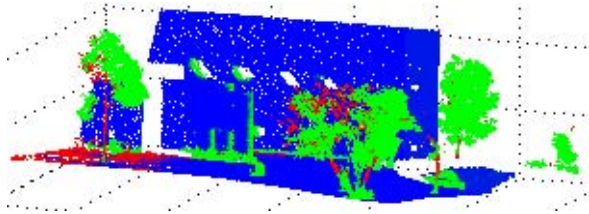


Figure 5: Sample result 2 with GMM (three Gaussians: red and blue for planar data class and green for cluttered data class)

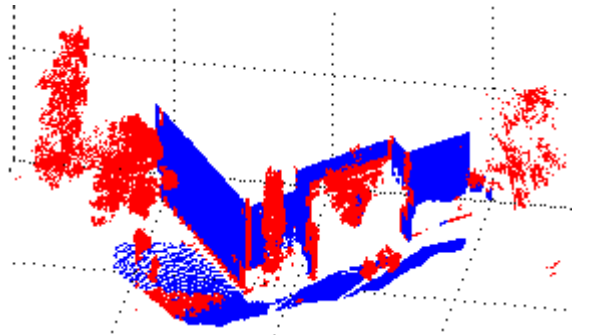


Figure 6: Sample result 1 of CRF with logarithm of saliency features

The result by fitting a Gaussian Mixture Model using Expectation Maximization algorithm yields a

classification accuracy of 60% to 70%. The major problem is that the values of the eigenvalues can vary in different support region size and density but the relationship among the eigenvalues remains for different data types. With a Conditional Random

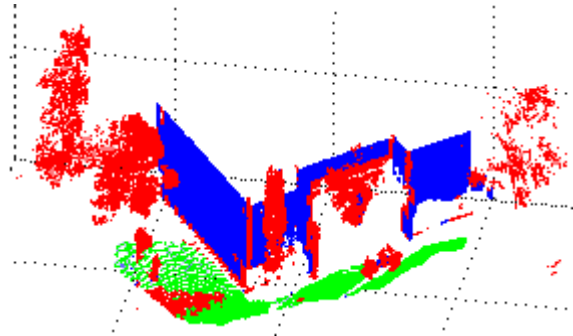


Figure 7: Sample result 1 of CRF with further terrain and building segmentation

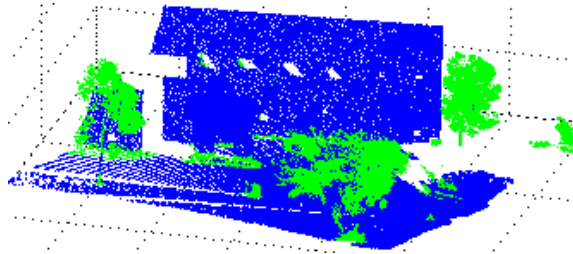


Figure 8: Sample result 2 of CRF with logarithm of saliency features

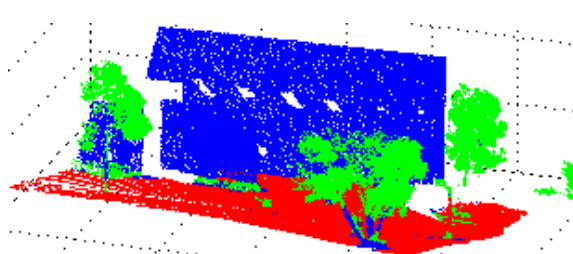


Figure 9: Sample result 2 of CRF with further terrain and building segmentation

Field as the training model, the result is promising: the misclassifications at the sparse density area are improved by 10% to 20%.

5 Conclusion

We have presented a method to perform automatic 3D data segmentation for terrain and building extraction in a vegetated environment. To summarize, our method uses a CRF trained on features extracted from the LIDAR data. The features include the normal vectors, normalised height and saliency features.

This approach is validated using data from a terrestrial Riegl laser scanner. On-going work includes point clustering via checking each point

against its neighbours for co-normality and coplanarity. We are also interested in implementing a CRF with adaptive k neighbouring points where the weight for neighbouring points can be adjusted based on the normals and the distance of the neighbouring points, and possibly introduce adaptive data reduction with the 3D scale theory.

6 Reference

- [1] A. Razdan, M. Tocheri, W. Sweitzer, and J. Rowe, "Adding semantics to 3D digital libraries," *Digital Libraries: People, Knowledge, and Technology. 5th International Conference on Asian Digital Libraries, ICADL 2002. Proceedings*, 2002.
- [2] I. Stamos and P. K. Allen, "Integration of range and image sensing for photo-realistic 3D modeling," *Proceedings 2000 ICRA*, vol.2, 2000.
- [3] R. Unnikrishnan and M. Hebert, "Robust extraction of multiple structures from non-uniformly sampled data," *IROS 2003*, vol.2, 2003.
- [4] J. F. Lalonde, R. Unnikrishnan, N. Vandapel, and M. Hebert, "Scale selection for classification of point-sampled 3D surfaces," *Proceedings. Fifth International Conference on 3-D Digital Imaging and Modeling*, 2005.
- [5] N. J. Mitra, A. Nguyen, and L. J. Guibas, "Estimating surface normals in noisy point cloud data," *Int. J. Comput. Geometry Appl.* 14, vol. (4-5), pp. 261-276, 2004.
- [6] L. Matikainen, J. Hyypä, and H. Hyypä, "Automatic detection of buildings from laser scanner data for map updating," *ISPRS Commission III.*, 2003.
- [7] D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Ng, "Discriminative learning of Markov random fields for segmentation of 3D scan data," *CVPR 2005*, II, 2005.
- [8] F. Rottensteiner, "Automatic generation of high-quality building models from lidar data," *IEEE Computer Graphics and Applications*, vol. 23, pp. 42-50, 2003.
- [9] P. Krishnamoorthy, K. L. Boyer, and P. J. Flynn, "Robust detection of buildings in digital surface models," *ICPR*, vol.1, 2002.
- [10] D. D. Lichti, "Spectral filtering and classification of terrestrial laser scanner point clouds," *Photogrammetric Record*, vol. 20, pp. 218-240, 2005.
- [11] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," *Comput. Graph. (USA)*, 26, 1992.
- [12] T. K. Dey and S. Goswami, "Provable surface reconstruction from noisy samples," *Proceedings of the Annual Symposium on Computational Geometry*, 2004.
- [13] T. K. Dey, G. Li, and J. Sun, "Normal estimation for point clouds: A comparison study for a Voronoi based method," *Point-Based Graphics, 2005 - Eurographics/IEEE VGTC Symposium Proceedings*, 2005.
- [14] J. Lafferty, A. McCallum, and F. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," *Proc. 18th International Conf. on Machine Learning*, 2001.
- [15] D. F. Wolf, G. S. Sukhatme, D. Fox, and W. Burgard, "Autonomous Terrain Mapping and Classification Using Hidden Markov Models," *ICRA*, 2005.
- [16] V. Kolmogorov and R. Zabih, "What Energy Functions Can Be Minimized via Graph Cuts?," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 147-159, 2004.
- [17] R. Triebel, K. Kersting, and W. Burgard, "Robust 3D scan point classification using associative Markov networks," *ICRA*, 2006.
- [18] W. Li and A. McCallum, "Rapid development of hindi named entity recognition using conditional random fields and feature induction," *ACM Transactions on Asian Language Information Processing*, vol. 2, pp. 290-294, 2003.
- [19] K. Sanjiv and M. Hebert, "Discriminative random fields: a discriminative framework for contextual interaction in classification," *Proceedings Ninth IEEE International Conference on Computer Vision*, vol.2, 2003.
- [20] X. He, R. S. Zemel, and M. A. Carreira-Perpinan, "Multiscale conditional random fields for image labeling," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2, 2004.
- [21] D. H. Tran, T. H. Pham, K. Satou, and T. B. Ho, "Conditional random fields for predicting and analyzing histone occupancy, acetylation and methylation areas in DNA sequences," *Lecture Notes in Computer Science*, 3907 NCS, 2006.
- [22] D. Pinto, A. McCallum, X. Wei, and W. Bruce Croft, "Table Extraction Using Conditional Random Fields," *SIGIR Forum*, 2003.
- [23] Y. Qi, M. Szummer, and T. P. Minka, "Diagram structure recognition by Bayesian conditional random fields," *CVPR 2005*, II, 2005.
- [24] S. F. Chen and R. Rosenfeld, "A Gaussian prior for smoothing maximum entropy models," *Technical Report CMUCS-99-108, Carnegie Mellon University*, 1999.

Real-Time Interaction Techniques for Meshless Deformation Based on Shape Matching

Alex Henriques and Burkhard Wünsche

Graphics Group, Department of Computer Science, University of Auckland, New Zealand

Email: burkhard@cs.auckland.ac.nz

Abstract

Meshless deformation based on shape matching is a new technique for simulating deformable objects which handles point-based objects and does not need connectivity information. The technique has been first presented in 2005 and is of interest to all fields which require fast, stable simulations which do not need to be physically correct. In particular the technique seems very suitable for use in virtual surgery applications and highly interactive real-time environments such as computer games. However, in contrast to traditional physically simulations, virtual environments require more complex and intuitive real-time interaction paradigms in order to increase the look and feel of the simulation and the immersive experience. We introduce techniques for picking, pushing and cutting objects simulated using meshless deformation based on shape matching. All interactions can be performed in real time, are unconditionally stable, easy to integrate into 3D rendering and game engines, and are easy-to-use and intuitive.

Keywords: deformable modeling, real-time simulation, interaction techniques, shape matching, virtual environments

1 Introduction

Advances in graphics hardware and rendering techniques have made it possible to develop realistic real-time interactive virtual environments. Typical examples are computer games, applications in architecture and urban design and to some extent visualisation applications in science, engineering and medicine. Despite these advances most of these applications still use models based on rigid-body physics due to their simplicity, easy control, and the existence of readily available fast simulation libraries such as ODE [1].

In 2005 meshless deformation based on shape matching was introduced as a new technique for simulating deformable objects. The technique does not require connectivity information for objects, is fast, unconditionally stable, and has low memory requirements. Consequently the technique might be very suitable for use in virtual surgery applications and highly interactive real-time environments such as computer games.

In this paper we introduce efficient interaction techniques, i.e. picking, pushing and cutting, for use with objects simulated using meshless deformation based on shape matching. The techniques can also be applied to other simulation methods but are particularly suitable for meshless deformation because when correctly implemented the technique is unconditionally

stable. Furthermore since meshless deformation does not require connectivity information we do not have to worry about the geometry (e.g. triangle aspect ratio) of the mesh representing a deformable object, and we can use models represented by point clouds. Consequently all interaction techniques can be executed in real time and can be easily integrated into a traditional 3D rendering or game engine.

Section 2 introduces the meshless deformation technique in more detail, section 3 describes the interaction techniques available in the application we have developed. Finally, section 4 summarises our results, and section 5 concludes.

2 Meshless Deformation

“Meshless Deformations Based on Shape Matching” is a recently developed technique for dynamically simulating deformable objects [2]. The underlying model is geometrically, as opposed to physically, motivated. It is unconditionally stable, does not require any pre-processing, and is simple to compute.

2.1 The Technique

Meshless deformation treats each object as a *point cloud*, or set of points, with no connectivity information required. To understand the basic idea,

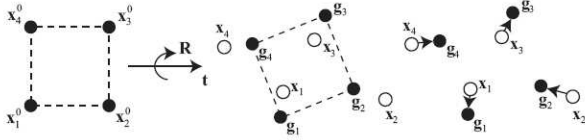


Figure 1: First, the original shape \mathbf{x}_i^0 is matched to the deformed shape \mathbf{x}_i . Then, the deformed points \mathbf{x}_i are pulled towards the matched shape \mathbf{g}_i (adapted from [2]).

let the initial configuration of points be \mathbf{x}_i^0 , and the deformed configuration of points at some later time be \mathbf{x}_i . As a set of unconnected particles, each \mathbf{x}_i responds to gravity and collisions, but no force acts to retain the overall object's shape. Meshless deformation's solution is to take the initial configuration \mathbf{x}_i^0 , then move and rotate it as closely as possible onto the actual configuration \mathbf{x}_i (see Figure 1). The rotated version of the initial configuration is now the set of *goal positions* \mathbf{g}_i which minimise the least squares distance to actual positions. Each particle is pulled towards its goal position after each time step, retaining the object's initial shape.

The fundamental equation that finds the optimal transformation from \mathbf{x}_i^0 to \mathbf{g}_i is that of "absolute orientation": given coordinates of a set of points as measured in two different Cartesian coordinate systems, find the optimal transformation between them [3]. To find this optimal transformation, the following sum is minimised.

$$\sum_i w_i (\mathbf{R}(\mathbf{x}_i^0 - \mathbf{t}_0) + \mathbf{t} - \mathbf{x}_i)^2$$

where \mathbf{R} is a pure rotation matrix. In meshless deformation, \mathbf{t}_0 is the centre of mass of the initial configuration, and \mathbf{t} is the centre of mass of the actual configuration. Müller et al. extend this equation by adding linear and quadratic matching; \mathbf{R} is replaced by a linear deformation matrix \mathbf{A} , or a quadratic deformation matrix $\tilde{\mathbf{A}}$. Thus, the goal positions can be not only a rotated version of the initial configuration, but a stretched, sheared, bent and twisted version. To produce a tendency towards the original undeformed state in linear and quadratic matching, \mathbf{R} is combined with \mathbf{A} or $\tilde{\mathbf{A}}$ to produce a final deformation matrix \mathbf{F} .

$$\mathbf{F} = \beta \tilde{\mathbf{A}} + (1 - \beta) \mathbf{R}$$

where β is a user defined constant between 0 and 1. Low β indicates a tendency mostly towards the rigid matched state, while high β indicates a tendency towards the quadratic match. The last

important constant is α , which defines the proportional distance the points move towards their goal positions \mathbf{g}_i every time step. When $\alpha = 1$, each point moves precisely to its goal position.

In summary, meshless deformation effectively transforms the original object by a matrix representing stretch, shear, bend and twist to find the closest match to the deformed object, then pulls the deformed object towards the goal positions represented by the transformed object.

2.2 Clusters

The primary disadvantage of meshless deformation is that goal positions are calculated by transforming the object with at best a quadratic deformation matrix, hence only 27 deformation modes are possible. Physical expressiveness may seem high, but significant limitations become apparent for objects more complicated than cubes and beach balls. These limitations are with respect to higher order deformation and local deformation. Consider two common objects as examples. A slithering snake might have two bends in it, which requires at least cubic deformations during animation, so it cannot be deformed with global quadratic equations. As a second example consider a sweatshirt with a hood. When using global deformations raising or lowering the hood is impossible to perform without bending the entire object. Note that quadratic equations do not have a compact support, i.e. are non-zero virtually everywhere.

To extend meshless deformation for local and higher order deformation, Müller et al. divide the set of particles into overlapping clusters, each with its own deformation modes and matrix. An entity consisting of multiple interacting clusters has a much greater range of deformation than an entity consisting of only one cluster. The shortcomings of the original implementation and our improvements for obtaining more physically realistic simulations are discussed in [4].

2.3 Evaluation

The advantages of meshless deformation are clear: it is fast and very easy to set up and tweak. The primary disadvantage of meshless deformation is that modes of deformation are quite limited. Clustering increases freedom, but is generally only well suited to objects with a small number of subparts, each of which deform at most quadratically. The only way to model more complex objects like cloth is to divide them into many fine grained clusters. But this is extremely inefficient and not very accurate – methods like mass-spring systems would be more suitable.

3 Interaction Techniques

In order to make a virtual world more realistic it is necessary to enable the user to interact with objects in a believable manner. Simulating both the look and feel of materials increases realism and the immersive experience. Furthermore advanced interactions are required for many applications such as virtual surgery simulations. In this section we introduce techniques for picking, constraining, pushing and cutting objects simulated using meshless deformation based on shape matching.

3.1 Picking

The main function of the picking mode is to grab objects and manipulate them with a spring force. The user can press the left mouse button to grab an object vertex, then drag the mouse around to control the direction of the spring force acting on that vertex. The spring force acts towards the position of the cursor represented by a red sphere. When the user moves the mouse, the red sphere moves along a plane facing the user. The mouse wheel can move the red sphere away from (mouse wheel up) or towards (mouse wheel down) the user. This moves the red sphere's plane of movement away from or towards the user, while keeping the plane's normal unchanged.

While dragging a spring force around, the user can release the left mouse button to stop the force and release the spring. Alternatively, the user can click the right mouse button to lock the force (i.e. the red sphere) in place. The user can then move around, change modes, or create a new spring force, while the original spring force remains in position. This makes it easy to “fix” an object in a deformed position. An example is shown in figure 2. To remove a locked in spring force, the user can click and drag on the red sphere to regain control of it then release the left mouse button, or press a key to remove all spring forces from every object.

3.2 Pushing

The main function of the pushing mode is to move objects by pushing them. A solid sphere follows the user's cursor in the same manner as the red sphere of the active spring force does in the picking mode above. Any objects colliding with the sphere undergo collision response forces. This is designed to mimic the user pushing objects around with his hand.

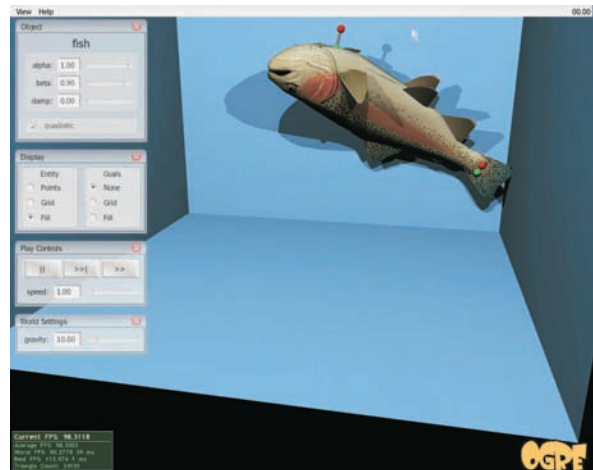


Figure 2: A deformed model of a trout fixed using two locked pick points.

3.3 Cutting

The main function of the cutting mode is to cut objects into separate pieces. The cursor turns into two cylinders designed to mimic a cutting instrument, e.g. a pair of scissors. To cut an object, the user moves the “scissors” to the appropriate position relative to the object, then holds down the left mouse button to begin the cutting process. The two “blades” of the scissors move closer together, and when they meet, every object the scissors intersect is severed along the plane of the scissors, creating two new separate objects.

To change the orientation of the scissors, the user can move the scissors towards him (mouse wheel down), away from him (mouse wheel up), or he can rotate the scissors about the y axis by holding down shift and dragging the left mouse button up or down.

3.3.1 Cutting Implementation

The cutting tool splits an object along a plane defined by the orientation of the scissors-shaped cursor. This simplifies the general cutting problem somewhat, as (a) we do not have to deal with partial cuts, and (b) the internal surface revealed by the cuts is always planar.

First, we define a *sever* operation which, taking an object o and a cutting plane c , removes all of o in c 's positive halfspace and neatly seals up the exposed cross-section. The *cut* operation then consists of two *sever* operations: $sever(o, c)$ and $sever(o_{clone}, -c)$, where o_{clone} is a clone of o and $-c$ is c with normal reversed.

The first step of *sever* is to separate o 's triangles into categories. Triangles with $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$

in the positive halfspace of the cutting plane are discarded. Triangles with $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ all in the negative halfspace of the cutting plane are kept. The remaining triangles straddle the cutting plane, and are cut along c to obtain a clean edge. These triangles have either exactly one or exactly two vertices in c 's negative halfspace. The former kind are shortened to produce the clean edge; the latter kind are cut to form two subtriangles (see Figure 3).

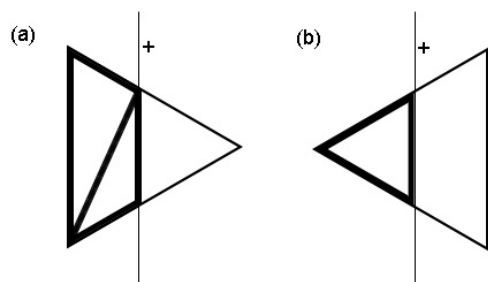


Figure 3: Triangles are made flush with the cutting plane's surface by creating two smaller triangles (a) or by shortening triangle edges (b).



Figure 4: Triangulation for a cut down the centre of an object shown as overlay (left) and around a diagonal cutting plane (right).

When this process is carried out over every triangle, a neat edge aligned with the cutting plane is produced. Figure 4 shows the results for an axis-aligned cutting plane (left) and for a diagonal cutting plane (right).

The next step is to seal the exposed cross-section. A surface is created by triangulating the newly created vertices touching the cutting plane with a Delaunay triangulation algorithm (see Figure 5). The triangles tend to be irregularly shaped because only vertices around the edge of the surface are fed into the algorithm. With no vertices in the centre, each triangle needs to span edge to edge. An improvement to our method would add new vertices inside the edges before running the Delaunay triangulation algorithm, resulting in more consistently sized and shaped triangles.

After triangulation is performed, the object is tetrahedralised and divided into clusters again.

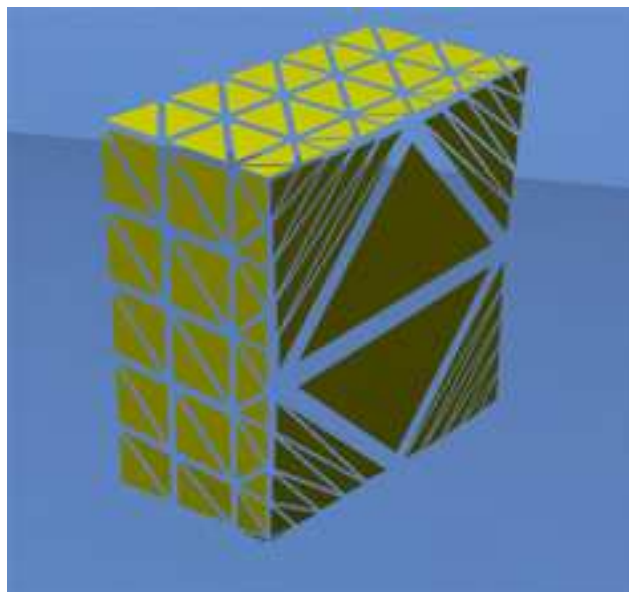


Figure 5: After a cut, the exposed internal hole is sealed up with a Delaunay triangulation.

Another possibility would be to keep what was left of the old tetrahedrons and clusters, but possible cluster degeneracy would need to be dealt with.

3.3.2 Problems

A requirement of the Delaunay triangulation algorithm is that the input is free of duplicate vertices (i.e. vertices with near identical positions). To achieve this, we simply create a separate list of duplicated cutting surface vertices, ensuring every new vertex added to the list has a unique position. The Delaunay triangulation is performed on this separate list. The edges of the surface produced, consisting of duplicate vertices, are thus sharp. This is desirable in most cutting applications. A rough cut could be easily achieved by displacing vertices along the cutting plane with a noise-based fractal function.

Since we use a 2D Delaunay triangulation algorithm we converted the 3D coordinates of the vertices of the cutting plane into 2D coordinates within this plane. The triangle vertex indices resulting from the algorithm can then be used to index the original 3D vertices. Unfortunately the particular implementation we used required that the input 2D points be sorted in order of increasing x . To preserve the mapping from 2D to 3D we created a data structure consisting of a 2D coordinate and an index into the 3D vertices' array, where the 3D vertex indexed is mapped to the 2D coordinate. The array of these data structures is then sorted into order of increasing x . The triangle resulting from the Delaunay

triangulation index into this array, from which we can extract the correct index into the 3D vertex array.

3.4 Collision

Several types of methods are available for detecting and responding to collisions between deformable objects. These include bounded volume hierarchies, stochastic methods, distance fields, spatial subdivision, and image-space techniques [5].

Our application uses spatial hashing [6] and penetration depth estimation [7] techniques. We found that collision detection was a performance bottleneck however. No “best way” to perform collision detection for deformable objects has been decided on yet, and future research will improve this area.

4 Results

We have implemented a meshless deformation algorithm based on shape matching and developed a test bed for simulation applications and interaction techniques [4] based on the Ogre 3D graphics engine [8]. The user can pick, push or cut deformable objects in real-time. Simple objects with limited modes of deformation are simulated best. Objects composed of simple subcomponents are simulated well with clusters. Objects with a very high number of deformation modes, such as cloth, can not be simulated efficiently [9].

Usability. We found that all interaction techniques were intuitive and easy to use and that they significantly increased user satisfaction (enjoyment) when interacting with the virtual environment. This is a strong indication that the implemented techniques are a useful addition to highly-interactive immersive environments although more formal tests are necessary to confirm this observation. The pick application works best for objects which deform globally, such as the trout shown in figure 2, whereas simulating locally deformable objects requires us to use multiple clusters as demonstrated in figure 6. The cutting tool proved particularly popular with users and significantly increased the look and feel of interacting with 3D objects (see figure 7).

Ease of implementation. We found meshless deformation relatively easy to implement and integrate into the 3D rendering engine Ogre. There are only two main differences between current 3D engines and what is required for deformable object simulation. Firstly, rigid objects have static sharable meshes, while deformable objects require updates to individual vertex positions every time step on

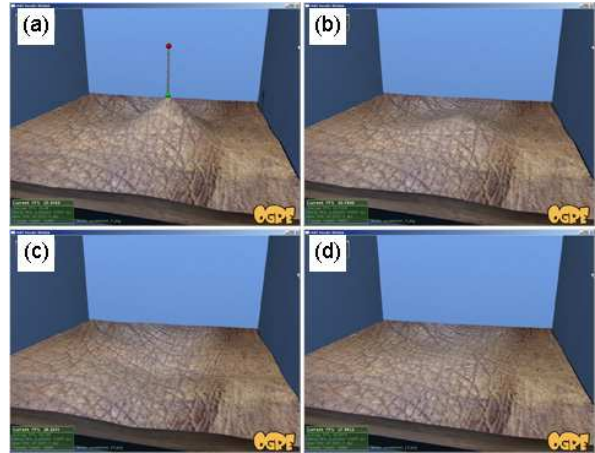


Figure 6: Behaviour of a 5×5 cluster skin patch in response to a user pick.

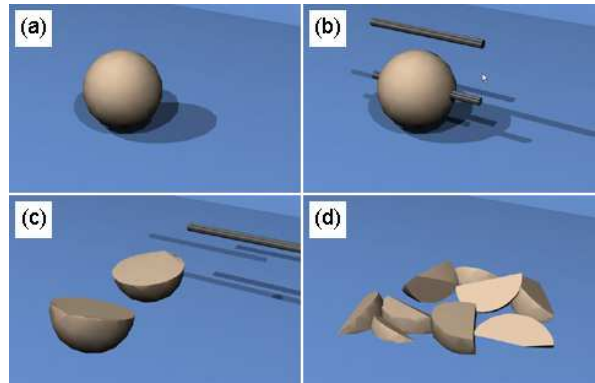


Figure 7: Cutting an object: (a) before cut, (b) during cut, (c) two resulting halves have rolled apart, (d) after further cuts.

their own mesh instance. Secondly, collision detection and response is a much slower, more difficult task for deformable objects.

Performance. Our environment is comparatively fast: We can simulate dozens of simple 32 tetrahedron objects with collisions in real-time and unconditional stability (see figure 8). Significantly better results could be achieved by optimising our algorithms and/or implementing them on the GPU.

Tweakability. The “gooeyness” and stiffness of each object can be easily modified using the α and β parameters. Further collision-response parameters can also be tweaked. The strength of surface area preservation can be specified with a force response curve. Volume preservation is automatic, but can be adapted to use a force response curve as well.

Disadvantages. The primary disadvantage of our environment is the lack of robust local deformation. For complex virtual surgery applications which of-



Figure 8: Large scale simulation of deformable objects.

ten require plausible localised deformation of an arbitrary region, our environment is less suitable. Also, even when simulation is visually plausible, it is usually not physically accurate.

5 Conclusion

We have implemented an improved algorithm for meshless deformation based on shape matching and we have presented several novel techniques to interact with these objects in a realistic and intuitive way. All interactions are performed in real time, are unconditionally stable and easy to integrate into 3D rendering and game engines. Informal user studies suggested that all interaction techniques significantly increase user satisfaction (enjoyment) when interacting with the virtual environment. Cutting could also easily be adapted to serve as a fracturing implementation.

Disadvantages are that performing local deformations requires models with sufficiently small clusters which is often not efficient. Also more improvements are necessary in order to apply our techniques to large scale objects and scenes. The cut operation so far can only perform full cuts and does not support local incisions which would be useful for a virtual surgery application or games where the player might want to slash an opponent.

In summary we believe that the implemented techniques are a useful addition to many highly-interactive immersive environments where speed and a more immersive feel are required but physical accuracy is not important.

6 Future Work

When cutting an object many new triangles are created along the cutting plane. Currently we give

each new triangle unique vertices which can result in an uneven look when using different vertex normals. In future we intend to utilise a hash table for vertices and normals similar to the one introduced by Wyvill et al. [10].

References

- [1] “Open Dynamics Engine home page.” <http://www.ode.org>.
- [2] M. Müller, B. Heidelberger, M. Teschner, and M. Gross, “Meshless deformations based on shape matching,” *ACM Trans. Graph.*, vol. 24, no. 3, pp. 471–478, 2005.
- [3] B. Horn, “Closed-form solution of absolute orientation using unit quaternions,” *Journal of the Optical Society of America A*, vol. 4, no. 4, pp. 629–642, 1987.
- [4] A. Henriques, “Meshless deformation for real-time soft tissue simulation,” BSc Honours dissertation, University of Auckland, 2006. (to be published in Oct 2006).
- [5] M. Teschner, S. Kimmerle, G. Zachmann, B. Heidelberger, L. Raghupathi, A. Fuhrmann, M.-P. Cani, F. Faure, N. Magnetat-Thalmann, and W. Strasser, “Collision detection for deformable objects,” in *Eurographics State-of-the-Art Report (EG-STAR)*, pp. 119–139, Eurographics Association, Eurographics Association, 2004. <http://www-evasion.imag.fr/Publications/2004/TKZHRFCFMS04>.
- [6] M. Teschner, B. Heidelberger, M. Mueller, D. Pomeranets, and M. Gross, “Optimized spatial hashing for collision detection of deformable objects,” 2003.
- [7] B. Heidelberger, M. Teschner, R. Keiser, M. Muller, and M. Gross, “Consistent penetration depth estimation for deformable collision response,” *Proceedings of Vision, Modeling, Visualization VMV04, Stanford, USA*, pp. 339–346, 2004.
- [8] “OGRE 3D home page.” <http://www.ogre3d.org>.
- [9] J. Rubin, “A framework for interactive and physically realistic cloth simulation,” 780 project report, University of Auckland, Feb. 2006. http://www.cs.auckland.ac.nz/~burkhard/Reports/2005_SS_JonathanRubin.pdf.
- [10] G. Wyvill, C. McPheeters, and B. Wyvill, “Animating soft objects,” *The Visual Computer*, vol. 2, pp. 235 – 242, Aug. 1986.

Terrain Reconstruction using LADAR and Optical Sensor Data from an Unmanned Air Vehicle

D. Gibbins¹, L. Swierkowski², P. Roberts¹ and A. Finn²

¹Sensor Signal Processing Group, Department of Electrical & Electronic Engineering,
The University of Adelaide, Australia.

² Defence Science & Technology Organisation (DSTO) Australia, Edinburgh, Australia,
Email: danny@eleceng.adelaide.edu.au, Leszek.Swierkowski@dsto.defence.gov.au,
proberts@eleceng.adelaide.edu.au

Abstract

The 3D reconstruction of terrain overflown by an unmanned air vehicle (UAV) has applications in mapping and area surveillance. As part of DSTO Australia's Automated Battle-Space Initiative (ABSI), DSTO, Tenix and Adelaide University are jointly working on a DSTO sponsored project to develop hardware and software systems to perform 3D terrain reconstruction from an aerial platform. This reconstruction can be achieved using a combination of GPS, attitude, optical and laser range finding sensors. However the reconstruction quality is dependent on compensating for sensor pose errors caused by platform vibration. This paper presents our ongoing work to develop a cost minimisation technique for 3D reconstruction that improves alignment using constraints from both the scanning laser range finder and registration terms derived from imagery from a co-located optical camera. Simulated and real ground test results are presented to support the approach developed so far.

Keywords: 3D modelling, terrain reconstruction, ladar, image registration, sensor fusion

1 Introduction

The 3D reconstruction of terrain overflown by an aircraft or small unmanned air vehicle (UAV) such as that illustrated in figure 1, has applications in mapping, target identification, and navigation. As part of DSTO Australia's Automated Battle-Space Initiative (ABSI), DSTO, Tenix and Adelaide University are jointly working on a DSTO funded project to develop hardware and software systems to perform 3D terrain reconstruction using a sensor payload mounted on a small low-cost unmanned air-vehicle.

Terrain reconstruction in 3D can be achieved in a number of ways. In the scenario discussed here, a Tenix designed sensor payload mounted on a UAV combining a GPS receiver, attitude sensor, a scanning laser range finder (Ladar) and an optical camera has been developed. The payload of co-located sensors is flown over the area of interest to produce a strip of GPS, attitude and ladar data which is then processed to form a 3D terrain estimate.

This combination of sensors enables individual range estimates to be directly converted to 3D spatial points by fusing the sensor data and applying simple geometric constraints. In short, the ladar range measurement $r(t)$ at time t to 3D coordinate mapping $X(t)$ can be written as:



Figure 1: An Australian built Aerosonde UAV mounted for a car assisted launch. The sensor payload (currently under construction by Tenix Defence) is designed to fit in the front section of the fuselage.

$$X(t) = R_s(t)R_lR_u(t) \begin{bmatrix} r(t) \\ 0 \\ 0 \end{bmatrix} + P_u(t) \quad (1)$$

where $R_u(t)$ and $P_u(t)$ are the rotation matrix and position vector for the UAV's pose in space, R_l describes the rotation of the ladar sensor relative to the UAV (known a-priori), and $R_s(t)$ is the rotation defining the direction of scan of the ladar sensor relative to R_l at time t . Thus given time stamped estimates of r, R_u, R_s and P_u from the GPS, attitude and ladar sensors a 3D terrain estimate can be constructed. An illustration is given in figure 2.

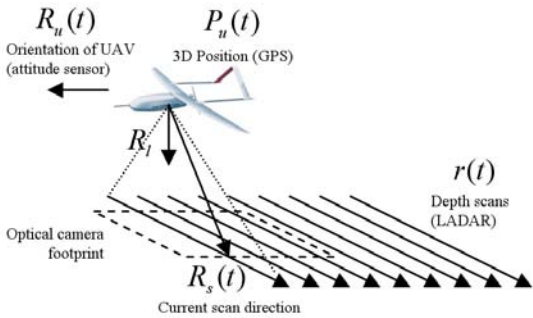


Figure 2: An illustration of the UAV geometries.

One problem with this simple geometric estimation of 3D points is that the quality of the reconstruction will be highly dependent on the level of errors in the estimates of R_u and P_u describing the UAV's pose, caused by platform vibration, sensor drift and timing errors during sensor fusion. For example vibrations in the UAV platform may cause errors in attitude estimation and hence result in discontinuities in the reconstructed surface, whilst sensor drift will produce low-frequency distortions in the reconstructed surface.

One possible solution to factoring out attitude sensors errors is to take advantage of a co-located camera sensor on board the UAV compare the observed scene motion to the motion estimates from the attitude and GPS sensors. In principle, if the ladar alignment is correct changes in ladar alignment should be mirrored by observable variations in the optical registration of the camera images from frame to frame.

In this paper, we present a summary of our progress towards developing an alignment scheme based on this observation which balances ladar scan registration constraints with those imposed by the observable change in UAV pose detected by a co-located camera. We begin by describing the underlying approach used by our work for ladar alignment without camera cues. This approach is motivated by the work of Thrun[1] and uses a cost minimisation strategy to align one ladar scan to the next. We then described how this approach can be extended to incorporate image registration constraints to remove short term UAV roll and shift errors. Results using simulated and real data are then presented to illustrate the approach.

2 Previous Work

The general problem of aligning 3D surface data has been examined by numerous research groups [2, 3, 4]. Whilst these approaches look at the problem of aligning two or more 3D data sets together, the problem considered here is not well suited to this as the sensor platform is moving rapidly in

one direction and earlier depth scans may bear no real relation to the depth data of the current scan. Consequently only some form of scan-by-scan alignment of the ladar data is feasible.

The problem of scan-by-scan alignment of ladar data for ground vehicles has been examined previously by Rofer[4]. In [4] the scan processing involves aligning the data based on histograms of surface orientation estimates extracted from the scan data. In the case of environments comprised of large flat surfaces the histograms will contain peaks that should agree from scan to scan if the data is properly aligned.

Another more relevant solution applied to helicopter platform data by Thrun[1] uses a cost minimisation function to balance scan alignment with the expected error characteristics of the sensor payload. This latter technique is the motivation behind the method presented in this paper. Here the technique has been extended to incorporate information coming from a co-located image sensor.

3 Reconstruction and Alignment

In Thrun[1], ladar scans collected using an unmanned helicopter moving at a steady velocity were re-aligned by registering the ladar scan in all 6 degrees of freedom using a negative log-likelihood cost function. One difficulty with the alignment step is that errors in UAV heading and pitch cannot be readily detected and resolved using the ladar data alone and consequently full 3D alignment is extremely difficult.

The approach taken here is a partial simplification of [1] which projects the current and previous scans onto a common 2D plane perpendicular to the direction of travel. If the flight is essentially a straight line this projection simplifies the geometry without significant loss of data. The 2D alignment problem is then solved using an appropriate cost function (see 3.1) that applies the required in plane rotation and shifts to align the new scan to the previous one. The aligned scan is then projected back into 3D space and the process repeated with the next scan.

3.1 Ladar-Only Reconstruction

Let Z_t represent the vector of 3D reconstruction estimates $X_t^{(1)}, X_t^{(2)}, X_t^{(3)}, \dots, X_t^{(n)}$ for the current scan of the laser range finder (ladar) and let Z_{t-1} represent the previous scan data (assumed to be correctly aligned). If f is a transformation function which applies a given alignment correction c to a given ladar scan Z , the optimal alignment between

two scans at times t and $t + 1$ should minimise the following condition with respect to some set of corrections c_t :

$$\sum (Z_{t-1} - f(Z_t, c_t)) \quad (2)$$

Assuming that the dominant errors are in roll, altitude and sideways translation of the UAV, the Z_t and Z_{t-1} terms can be approximated by Z'_t and Z'_{t-1} where Z' denotes the 2D projection of the two scans onto a plane perpendicular to the current direction of travel. In this case f reduces to a linear-conformal transformation and two shift terms (described by c_t).

This constraint by itself is not sufficient to produce a stable solution to the alignment problem[1], nor does it include any a-priori information about the nature of the measurement errors. Ideally the orientation of the solution should not differ significantly from the initial estimate, nor should the correction terms vary wildly from one scan to the next. In other words, the best solution is one where equation (2) is minimised, whilst keeping the magnitude of c_t and $(c_t - c_{t-1})$ small. This leads us to propose the following cost function, which is minimised with respect to the correction term c_t :

$$F(Z'_t, Z'_{t-1}, c_t, c_{t-1}) = c_t^T A^{-1} c_t + (c_t - c_{t-1})^T D^{-1} (c_t - c_{t-1}) + (\min(\alpha, Z'_{t-1} - f(Z'_t, c_t)))^T B^{-1} (\min(\alpha, Z'_{t-1} - f(Z'_t, c_t))) \quad (3)$$

where c_t and c_{t-1} are the current and previous correction terms, $f(Z'_t, c_t)$ is the projection of the current scan Z'_t using the alignment correction term c_t and α is a threshold on the alignment error between the two scans. The remaining terms A , D and B are inverse covariance matrices. These control the allowable level of correction applied, the correction as a function of its rate of change with time, and the measurement errors from one scan to the next.

Overall this cost function attempts to strike a balance between alignment correction and the amount of variation likely to be present in the overflow terrain. The addition of the threshold term α to the alignment cost[1] is intended to reduce the impact on the cost functions of discontinuities in the ladar data caused when a new building first enters the ladar's field of view.

In practice the alignment scheme described by equation (3) relies on a good estimate of the alignment error between one scan and the next in the 2D projection. In Thrun[1] this is achieved by minimising the distance of each point in the old scan to each point in the new scan. However, this has been found to be computationally expensive

and does not correctly handle situations where the sample data is sparse. Instead, in the approach taken here, the points of the new scan are re-sampled to conform with the sample points in the previous scan (ie. if the 2D projection is described by a coordinate (x, z) where z is height, then the new scan is re-sampled onto the same x values as the previous scan using linear interpolation).

3.2 Ladar Reconstruction Incorporating Optical Registration Cues

In the case of a co-located optical camera, any corrections made to the ladar data must be consistent with observed changes in the registration of images from frame to frame. This has at least two consequences:

1. Any errors in heading will show up as rotational errors between the expected registration based on GPS and attitude measurements and the actual registration estimates computed using, say, optical flow[5].
2. Any errors in UAV roll, height or sideways shift will be reflected in inconsistencies in the sideways registration of the image data. For example, a ladar correction for UAV roll ought to be observable as a sideways shift in the image registration data (ie. along the image x -axis) beyond that predicted by the attitude and GPS sensors alone.

The relationship between image registration and ladar alignment is also summarised in figure 3.

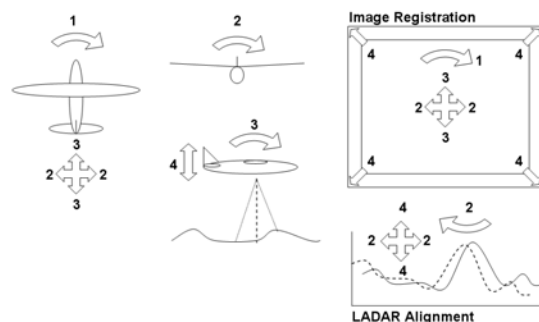


Figure 3: An illustration of the relationship between ladar and image registration. In this work only the roll and sideways shift relations are considered.

In the scheme being developed the first observation is used to correct for heading estimation errors prior to 3D estimation, and the adjusted 3D estimates are then aligned taking the second constraint into account. Incorporating the image registration constraint in 3, results in a new cost function, again minimised with respect to c_t :

$$\begin{aligned}
F(Z'_t, Z'_{t-1}, c_t, c_{t-1}) = & c_t^T A^{-1} c_t + \\
& (c_t - c_{t-1})^T D^{-1} (c_t - c_{t-1}) + \\
& (\min(\alpha, z'_{t-1} - f(z'_t, c_t)))^T B^{-1} \\
& (\min(\alpha, z'_{t-1} - f(z'_t, c_t))) + \\
& (s_t^p(c_t) - s_t^e)^T E^{-1} (s_t^p(c_t) - s_t^e)
\end{aligned} \quad (4)$$

Here the new terms are s_t^p the predicted registration x -shift at time t given the correction term c_t and expected shift s_t^e based on image registration. As a result of this new term the alignment of ladar and camera imagery must be consistent with one another if the correction term is valid.

In practice, the estimated shift s_t^e can be computed by registering images from the co-located camera[5], whilst the predicted shift s_t^p can be estimated from the GPS and attitude sensor data prior to minimisation given knowledge of the camera system and assuming the correction term c_t can be converted into a change in camera pose. This is tractable if the 2D transformation is centred on the current camera/ladar position as the 2D rotation can be related to camera roll and the two shift terms to altitude and sideways shift of the UAV position.

4 Simulation Results

The two methods of approach presented in sections 3.1 and 3.2 have been assessed using a simulated terrain model of gable-roofed buildings. Simulated fly-overs and the associated ladar and optical data were then constructed and the GPS and attitude estimates intentionally distorted to simulate vibration and sensor drift. An illustration of the simulated terrain and sample optical imagery used in our initial tests is shown in figure 4.

4.1 Ladar-only Alignment Results

Figure 5 illustrates two typical reconstruction results before and after ladar-only alignment correction. Here the camera pose errors were in the order of 5 degrees in heading, pitch and roll, and 3 metres in height. The simulation itself represents a fly-over of simple rectangular buildings using a mini-UAV at a height of around 80 metres. What can be seen in the result is that the uncorrected surface has a series of ripples through it related to the sensor pose drift errors. The re-alignment result shown in the lower part of figure 5 contains significantly fewer of these errors. The remaining errors in the reconstruction relate to heading errors which cannot be compensated for by the alignment scheme.

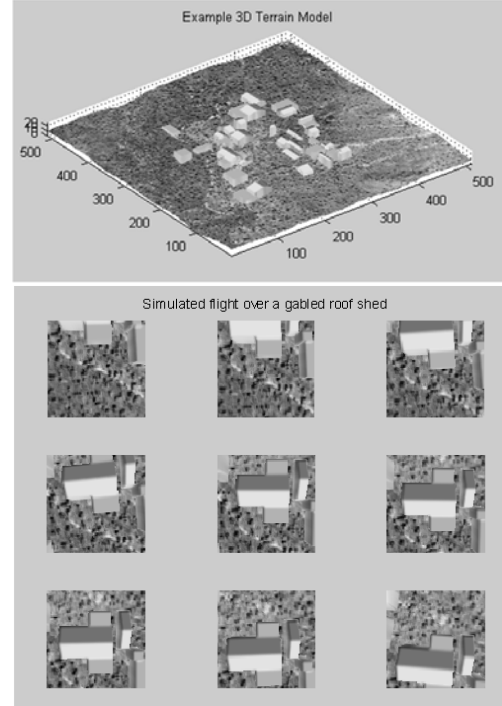


Figure 4: Simulated terrain model and flight imagery used in experiments

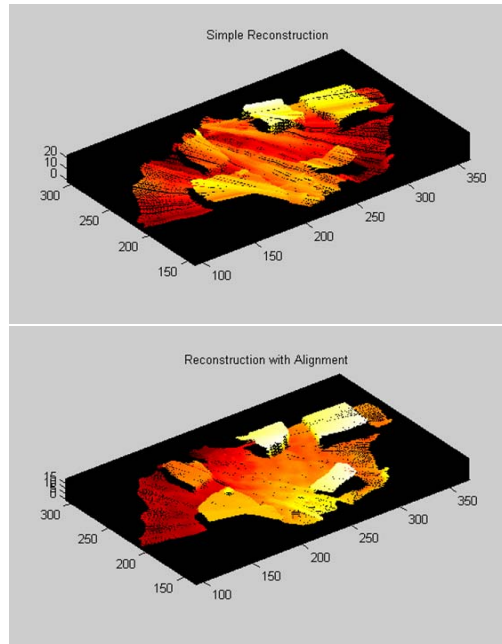


Figure 5: Simulated Result using Ladar data alone.

4.2 Ladar and Optical Alignment

Figure 6 shows a typical comparison of reconstruction using either raw, ladar-only or ladar and camera registration. Here the example has been further complicated by the combination of short term and long term sensor errors. In the case of the camera and ladar reconstruction, the central region of the result contains the grey-level image data associated with the ladar scans (the two sensors were

intentionally given different fields of view). What can be seen in this example is that the combination of ladar and optical imagery (lower result) has improved the reconstruction as compared to the ladar-only result (middle). In the case of the ladar only result, local minima in the cost function resulted in jumps in the alignment corrections which are not present when incorporating the optical information into the cost function.

In these examples, a pinhole camera model was employed to approximate the camera and an optical flow technique based on [5] was used to register the images.

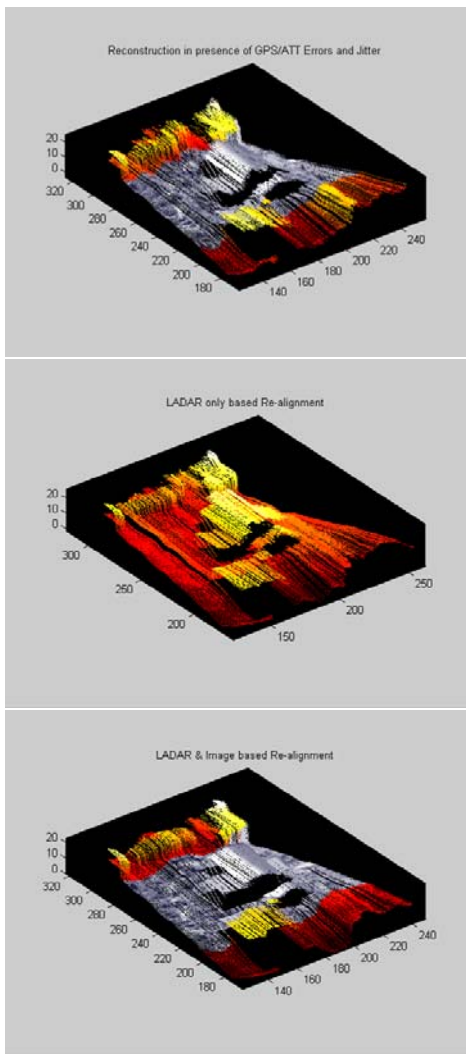


Figure 6: Comparison Simulation Result using Ladar and Optical data.

5 Initial Work Using Real Data

Prior to UAV data collection trials due later this year, the alignment scheme has been adapted to a series of ground based trials conducted in mid-June 2006. These trials consisted of mounting the prototype UAV payload to a 4-wheel drive vehicle which

was then driven past a series of buildings whilst a combination GPS, attitude, ladar and video samples were collected. The ladar scanner operates at around 75Hz (40 samples at 1 degree increments) and the optical, attitude and GPS sensors sample rates were 25, 25 and 1Hz respectively. For the purposes of reconstruction each of the above sensor outputs was tagged with a common time-stamp.

An example reconstruction of a series of factory sheds using this prototype equipment is given in figure 7 with a close up in figure 8. Here the recorded attitude data was intentionally jittered by around 3 degrees to simulate vibration effects and the ladar alignment scheme applied. What can be seen in this example is that the re-alignment has significantly improved the quality of the reconstruction.

Another example with the optical image data registered and overlaid over the 3D sample points is shown in figure 9.

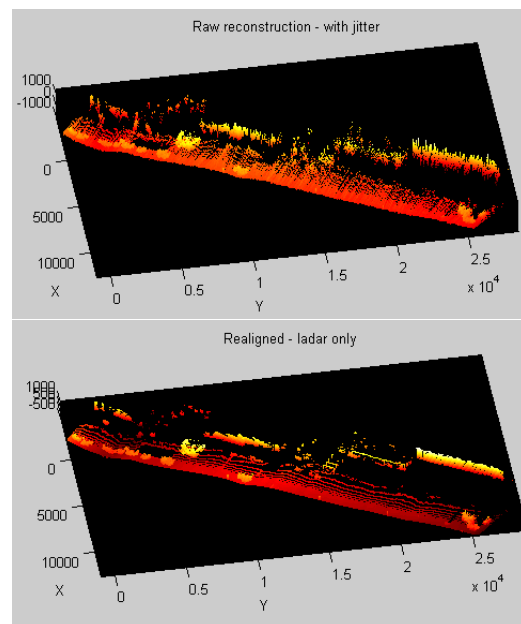


Figure 7: Real data reconstruction with jitter added before and after alignment. The data sequence represents a drive past a series of factory sheds.

6 Conclusions

This paper presents a method of ladar scan alignment designed to reduce the effects of GPS and attitude sensor errors in 3D terrain reconstructions. The proposed approach uses 2D projection of the 3D scans and a cost minimisation technique to find suitable alignments of the ladar scans. The approach can be further improved by taking advantage of visual cues from a co-located camera as demonstrated by the presented examples.

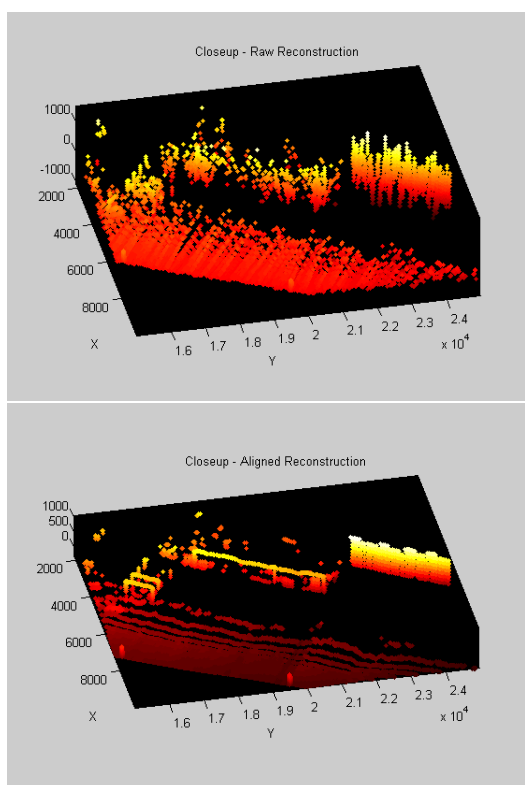


Figure 8: A close up of part of the real data reconstruction shown in figure 7.

7 Future Work

Future work will focus on applying the ladar and optical alignment techniques to real datasets collected using a UAV platform.

8 Acknowledgements

This project is supported by the Defence Science and Technology Organisation (DSTO) Australia and is funded under the Automated Battle-Space Initiative (ABSI).

The authors would like to thank the DSTO for supporting the work, and specifically the help of J.P. Gibard (DSTO), A. Bailey (DSTO) and the team at Tenix Defence (T.Depieri and M.Ziebarth) as well as many others without which this work would not have been possible.

References

- [1] S. Thrun, M. Diel, and D. Hahnel, "Scan alignment and 3-d surface modelling with a helicopter platform," in *Proceedings of the 4th International Conference on Field and Service Robots*, July 2003.
- [2] P. Besl and N. McKay, "A method of registration of 3-d shapes," *IEEE Trans. Pattern*

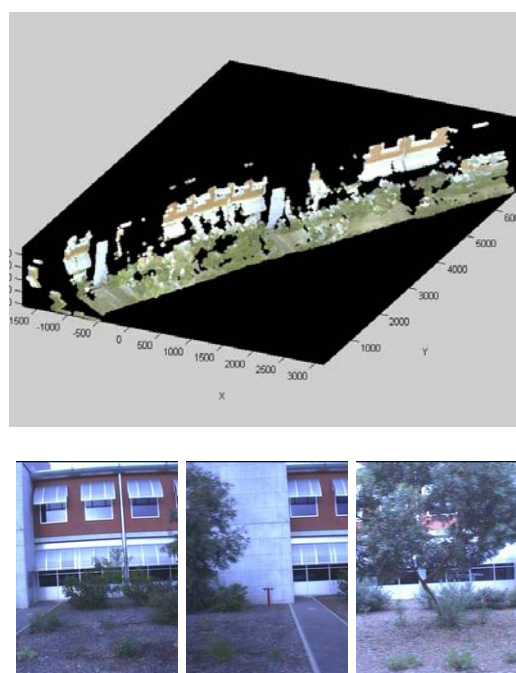


Figure 9: Real Data Reconstruction with optical image overlay. Here the ladar was mounted sideways on a 4WD vehicle. Three images from the sequence are shown for comparison.

Analysis and Machine Intelligence, vol. 14, pp. 239–256, February 1992.

- [3] S. Thrun, M. Diel, and D. Hahnel, "Map building with mobile robots in populated environments," 2001.
- [4] T. Rofer, "Using histogram correlation to create consistent laser scan maps," 2001.
- [5] T. Guatama and M. van Hulle, "A phase-based approach to the estimation of the optical flow field using spatial filtering," *IEEE Trans. Neural Networks*, vol. 13, no. 5, pp. 1127–1136, 2002.

Image processing of cryo-electron micrographs of helical crystals - 3D architecture of a novel bacterial appendage

J. Li¹, S. Manning¹, S. Turner¹, M. Kikkawa², and A.K. Mitra¹

¹ School of Biological Sciences, The University of Auckland.

² Department of Cell Biology, University of Texas Southwestern Medical Center.

Email: a.mitra@auckland.ac.nz

Abstract

This paper describes the determination of the 3D structure of a novel bacterial appendage, which naturally forms helical tubular crystals. For this purpose we have, using a transmission electron microscope, imaged such crystals when suspended in vitrified buffer and applied helical image processing methods to arrive at the 3D reconstruction. We describe the key steps in this processing: (1) Computation of the diffraction pattern, (2) Indexing of the diffraction pattern to determine the “selection rule”, (3) Accumulation of the Fourier components (big G) along layer lines and correction for the microscope contrast transfer function, (4) Calculation of the Fourier transform inversion (little g) of big G and finally (5) Calculation of reconstructed density by Fourier-Bessel inversion. We describe the architecture of the tubular crystals at $\sim 28\text{\AA}$ resolution.

Keywords: helical structure, Fourier transform, diffraction pattern, Fourier-Bessel inversion

1 Introduction

Many biological macromolecules in their functional state exist as polymers that are helical in nature. A helical assembly is the simplest arrangement of a repeating motif¹ in 3D, and when it is regularly arranged related by a simple rotation and translation along an axis (Figure 1), the assembly can be thought of as a 1D crystal (as opposed to 2D and 3D crystals). Helical assemblies can be imaged in a transmission electron microscope (TEM) and image processing can be carried out to reveal the 3D structure of the subunit. Such an analysis from helical crystals is particularly effective since, unlike in the case of 2D crystals, for instance of membrane proteins, tilting of sample in the microscope is not required and the resolution is isotropic due to the lack of a missing cone of data.

A strain of bacteria that was isolated from a wastewater treatment system and belongs to the *Acidovorax* genus has been noted to form biofilms. These bacteria produce a novel appendage ($\sim 55\text{nm}$ to $\sim 62\text{nm}$ diameter) that appear as sheaths enclosing a cargo of unknown chemical nature, frequently connected to neighbouring bacterial cells, and appear to be

involved in bacterial macro communication. These appendages display a natural helical crystal that is subject to helical image processing [1, 2, 3] for revealing the 3D architecture. We have recorded images of such tubular bacterial appendage (TBA) by trapping these in an unperturbed state in vitrified buffer. The objective of this work is to determine the 3D structure by application of classical image processing techniques.

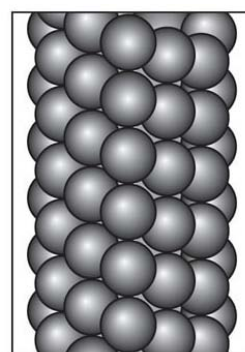


Figure 1: Left-handed helical model [4].

The paper is organized as follows. Section 2 considers the theoretical background of the main analysis. Section 3 presents the analyzing methods and the experimental results. A brief discussion of the results and conclusions follow in Section 4.

¹Motif: smallest biological unit comprising the regular assembly, e.g. a protein molecule.

2 Theoretical background

2.1 Diffraction pattern of a helical crystal

The first analysis of the Fourier transform of a helical object was developed by Cochran et al.[1] and Klug et al.[2]. The main equations of the Fourier transform of a helical object are described by Hawkes et al.[5]:

- The Fourier transform of a helical object can be written as:

$$F(R, \Phi, Z) = \sum_{n=-\infty}^{\infty} \exp\{in(\Phi + \pi/2)\} G_n(R, Z) \quad (1)$$

where

$$G_n(R, Z) = \frac{1}{2\pi i^n} \int_0^{2\pi} \exp(-in\Phi) F(R, \Phi, Z) d\Phi \quad (2)$$

- The Fourier-Bessel transform can be written as:

$$g_n(r, Z) = \int_0^{\infty} G_n(R, Z) J_n(2\pi Rr) 2\pi R dR \quad (3)$$

where J_n is a Bessel function corresponds with layer-plane n .

- The Fourier-Bessel inversion can be written as:

$$f(r, \Phi, Z) = \sum_{n=-\infty}^{\infty} \exp(in\phi) \int_{-\infty}^{\infty} g_n(r, Z) \exp(-2\pi i Z z) dZ \quad (4)$$

The diffraction pattern of a helical crystal is confined to a set of “layer lines” due to the regular repeat along the helix axis and the amplitude and phase of spots on the layer lines are defined by Bessel Functions of various orders. The reciprocal spacing of a given layer line corresponds to the inverse of an integral multiple of the repeat distance i.e. the translation along the helix axis that brings one motif to be in exact register with another motif. The Fourier transform of a helical particle consists of parallel planes (Figure 2), which appear as layer lines in the diffraction pattern because the Fourier transform of a micrograph is a central cross-section of the transform [3]. Thus it is possible to reconstruct the three-dimension structure of the tube based on a two-dimension micrograph.

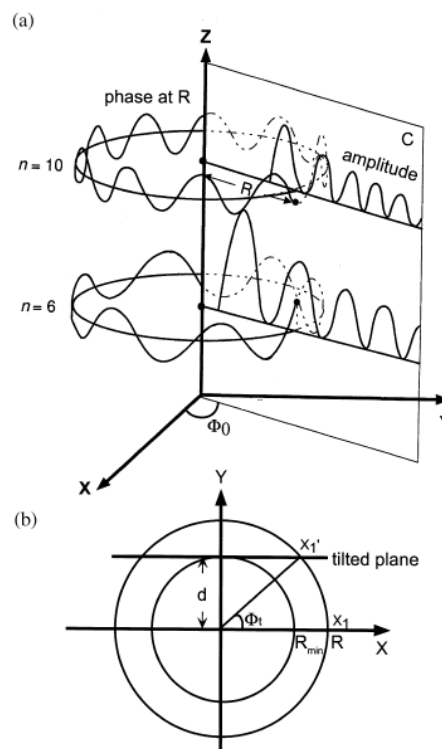


Figure 2: Three-dimensional Fourier transform of a helical particle and the relationship between layer-lines and layer-planes [3].

2.2 Selection rule

The indexing of the diffraction pattern entails assignment of (n, l) values i.e. the Bessel order contributing to the layer line and the layer line number. The diffraction pattern can be thought of as that corresponding to a 2D crystal (created by opening out the helix on a plane containing the helix axis), with the (n, l) values for the two primary vectors $(n_{1,0}, l_{1,0})$ and $(n_{0,1}, l_{0,1})$ necessary to index the complete lattice. This description is also called the selection rule and reflects the arrangement of the helix along its length (Z-axis). Diffraction patterns of images that agree to a given selection rule (i.e. the same $(n_{1,0}, l_{1,0})$ and $(n_{0,1}, l_{0,1})$) are called as belonging to the same helical family.

2.3 Contrast transfer function

Most images in TEM are recorded at various level of under focus to enhance the phase contrast. The Fourier component in the computed transform of the image is the product of the Fourier component in the object multiplied by the so-called CTF:

$$CTF(\lambda, g, \Delta f, C_s) = \begin{aligned} & -w_1 \sin[\chi(\lambda, g, \Delta f, C_s)] \\ & -w_2 \cos[\chi(\lambda, g, \Delta f, C_s)] \end{aligned} \quad (5)$$

with

$$\begin{aligned} \chi(\lambda, g, \Delta f, C_s) &= \pi \lambda g^2 (\Delta f - C_s) / 2 \lambda^2 g^2 \\ w_1 &= \sqrt{1 - A^2} \\ w_2 &= A \end{aligned} \quad (6)$$

where λ is the electron wavelength, g is the scattering vector describing the difference between the wave vectors of the unscattered and scattered electrons, C_s is the spherical aberration coefficient of the object lens, and A the percentage of amplitude contrast. The defocus Δf is given by [6]

$$\Delta f = \frac{1}{2} [DF_1 + DF_2 + (DF_1 - DF_2) \cos(2[\alpha_g - \alpha_{ast}])] \quad (7)$$

where DF_1 and DF_2 are the two defocus values describing the defocus in two perpendicular directions in an image when astigmatism is present, α_{ast} is the angel between the DF_1 and the X-axis, and α_g is the angel between the direction of the scattering vector g and the X-axis.

Contrast transfer function causes resolution-dependent amplitude modulations and phase reversals in the images [7]. The phase changes more rapidly in the higher defocus images and the amplitude oscillates faster towards the higher resolution as shown in Figure 3.

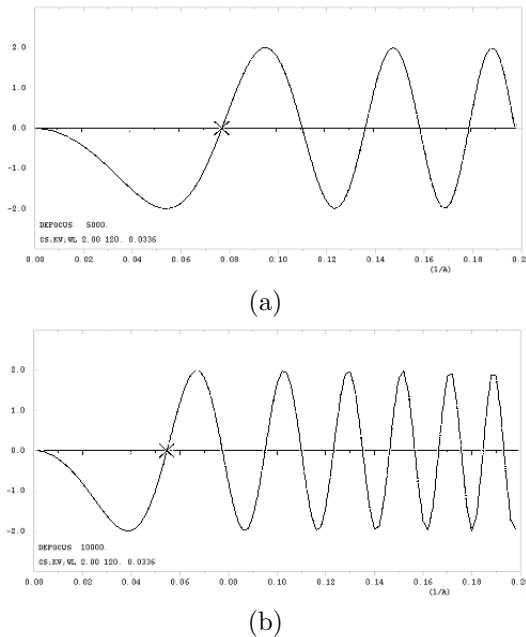


Figure 3: CTF plots: $C_s = 2.0mm$, voltage = $120kV$. (a) defocus= 5000\AA ; (b) defocus= 20000\AA .

3 Methods

The overall steps for the image processing of the helical tubes are described in the work-flow scheme shown in Figure 4.

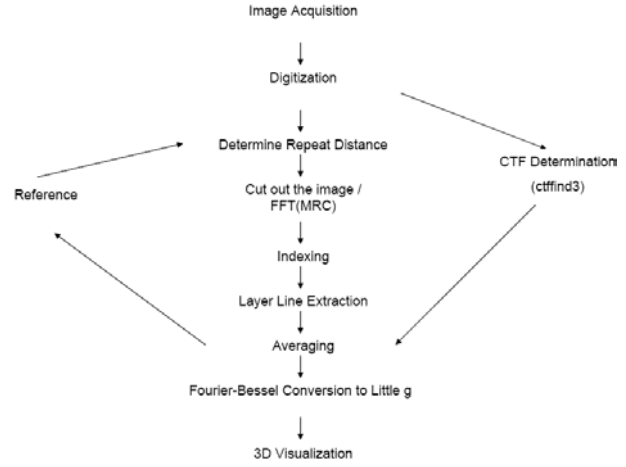


Figure 4: Work-flow.

3.1 Preliminary steps

The TBA specimens were applied on an EM grid covered with a “holey” film. After blotting excess solution the grid was rapidly plunged in a cryogen (liquid ethane) using a guillotine device. This process vitrified TBAs suspended in a layer of buffer within the holes, while preserving their cylindrical symmetry. Using a GATAN cryo-holder, the TBAs were examined at $-163^\circ C$ in a Tecnai12 TEM operated at 120kV and images recorded at a nominal magnification of 30,000. The micrographs were examined on an optical diffractometer to check for the optical quality. Selected regions of a micrograph was digitised using a Leafscan scanner at a raster step of $10\mu m$. The scanned image of a micrograph is shown in Figure 5.

The region of interest of the scanned image is selected based on two conditions. First of all, the region can include the whole helical tube or part of the helical tube. However, the diameter along the tube must be constant or very similar (for example ± 5 pixels), because the diameter affects the selection rule; also the tube must be continuous and straight. Secondly, the region is considered as one of interest only if its corresponding diffraction pattern is good. This means the diffraction pattern must have a reasonable number of strong layer lines, which are thin (1-pixel thick), long (at least 5-pixel long) and symmetric as well, as shown in Figure 6.

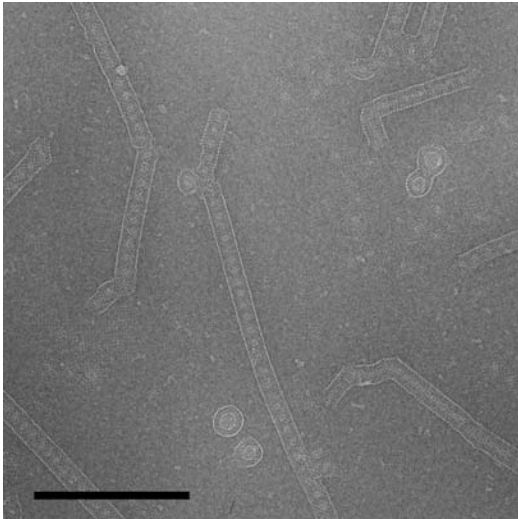


Figure 5: Scanned image of a micrograph. Scale bar represents 4000 Angstroms.

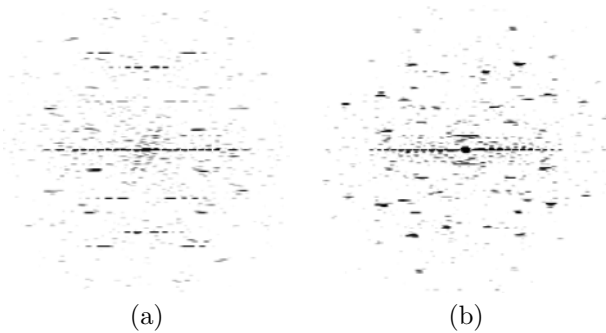


Figure 6: Diffraction pattern: (a) image 4815; (b) image 5534.

3.2 Find repeat distance

The helix repeats exactly when translated a distance c along its axis (Z-axis) [5]. The distance c is called the repeat distance of the helical tube. The first step of finding the repeat distance is to do the correlation of two separate boxed areas along the tube. Next, the rough repeat distance is determined by the parts of the tube that display best correlation. The rough repeat distance is refined later.

3.3 Indexing of diffraction patterns

Indexing of diffraction patterns is the process of fitting a lattice to the diffraction pattern, in order to find out the selection rule. The main steps of indexing the diffraction pattern are:

1. select two vectors for the lattice. Each vector is the direction to a strong layer line, which is very close to the meridian (Y-axis). There are two possible solution of the lattice: both

vectors belong to the same quadrant or fall into different quadrants,

2. fit the lattice on diffraction patterns: the lattice intersections must overlap on one layer line of each strong pair,
3. assign order (n, l) to each layer line. n is the Bessel order, calculated by $\frac{2\pi r R_M}{1024}$ where r is the diameter of the tube and R_m is the distance between the layer line center and the meridian. l is the layer line number, calculated by $\frac{yc}{1024}$ where y is the Y-coordinate of the layer line and c is the repeat distance. 1024 is the size of the diffraction pattern (1024 x 1024 pixels).

The most difficult part of the indexing is to determine the parity of n , which is based on the phase difference of the layer line pair. If the tube is very tilted out of the plane, the phase information will become unreliable. Also, finding the true repeat distance is hard in some case (i.e. very low contrast image leads to poor correlation). An example of indexing of a diffraction pattern of a helical tube is shown in Figure 7.

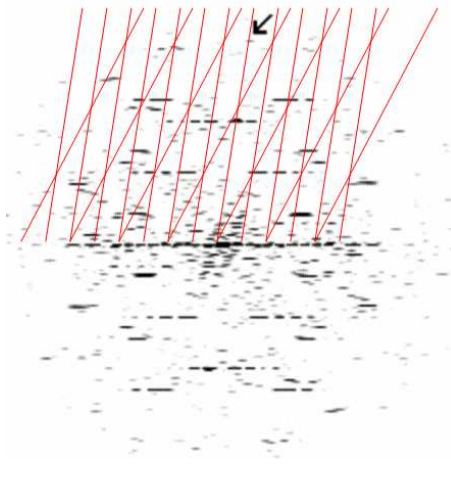


Figure 7: Indexing of image 4815.

3.4 Reconstruction process

Once the repeat distance and indexing are ascertained the reconstruction process is carried out using the Fourier Bessel algorithm described in section 2.1. For this purpose we apply the package developed by Unwin and colleagues [8] and by Kikkawa and colleagues [9]. The process includes layer lines extraction, averaging data sets, refinement of repeat distance and Fourier-Bessel inversion.

3.4.1 Contrast transfer function correction

Contrast transfer function (CTF) correction can improve the signal-to-noise ratio of the final reconstruction, by correcting for phase inversion. At the same time the background noise is filtered by subtracting the background in the power spectrum. The defocus parameters of the EM images in Equation (7) can be obtained by e.g. using the computer program CTFFIND3 [7].

3.5 Results from reconstruction

Some three-dimensional reconstructions of helical tubes are shown in Figure 8 and Figure 9. These include reconstructions before and after CTF correction. The results show that CTF correction can improve the signal-to-noise ration, because the sub units of the structures with CTF correction are much clearer than those of the structures without CTF correction.

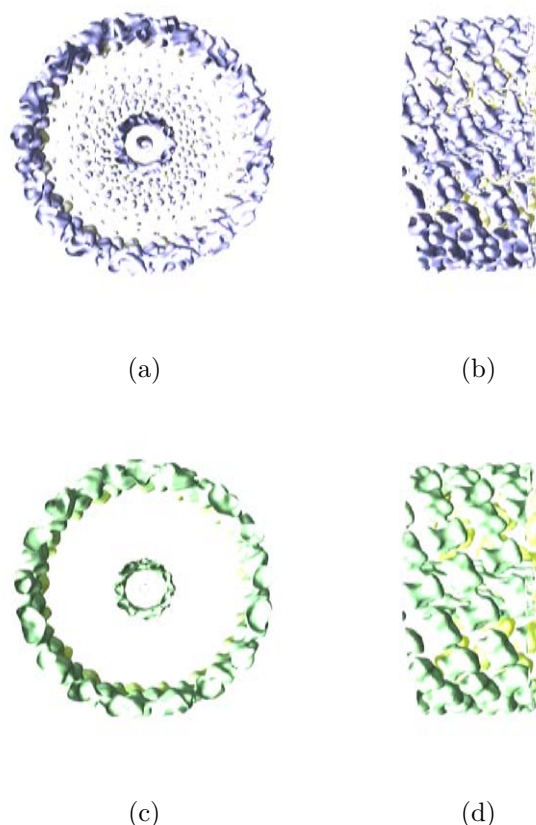


Figure 8: Side views of image 4815. (a), (b): without CTF correction; (c), (d): after CTF correction).

Since there appears to be a considerable variation in the type of helical families, direct Fourier

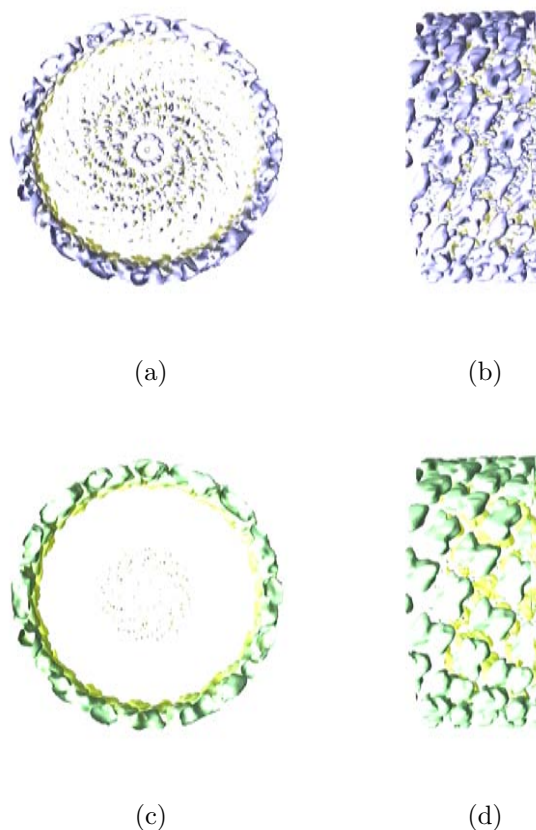


Figure 9: Side views of image 5534. (a), (b): without CTF correction; (c), (d): after CTF correction).

averaging of data from many tubes was not possible. The repeating subunit revealed at a resolution of about 28\AA appears to enclose a molecular mass of about 250kD . Biochemistry of the TBA shows that the major protein component has a mass of 63kD , therefore each subunit is a dimer² of two protein monomers. The internal core appears as a stalk of about 50\AA radius.

There are no published 3D structure of the subunit comprising the helical bacterial appendage that we have studied. We are carrying out reconstruction from a large number of images, many of them belonging to distinct helical families. The accuracy of the analysis is judged from the self consistency of the 3D structure of individual subunit in these reconstructions (e.g. compare subunits illustrated in Figure 8 and Figure 9.)

²Dimer: a pair of protein molecules, each called a monomer.

4 Conclusions

In this paper, we have presented an application for the processing of low-contrast cryo-electron micrographs of helical structures to reveal 3D architecture of a novel bacterial appendage. Processing of additional images and real space averaging to improve the details of the subunit is needed. However, our analysis has started revealing details of this novel structure that may help understand its biological function.

5 Acknowledgements

This work was supported by grants from the National Institutes of Health, USA to A. K.M and M.K.

References

- [1] W. Cochran, F. H. C. Crick, and V. Vand, "The structure of synthetic polypeptides. i. the transform of atoms on a helix," *Acta Cryst.*, no. 5, pp. 581–586, 1952.
- [2] A. Klug, F. H. C. Crick, and H. W. Wyckoff, "Diffraction of helical structures," *Acta Cryst.*, no. 11, pp. 199–213, 1958.
- [3] C. Toyoshima, "Structure determination of tubular crystals of membrane proteins. I. Indexing of diffraction patterns," *Ultramicroscopy*, no. 84, pp. 1–14, 2000.
- [4] Blackwell Microbiology, "Home page." <http://www.blackwellpublishing.com>.
- [5] P. W. Hawkes and U. Valdre, *Biophysical Electron Microscopy - Basic Concepts and Modern Techniques*. Academic Press, 1990.
- [6] R. Henderson, J. M. Baldwin, K. H. Downing, J. Lepault, and F. Zemlin, "Structure of purple membrane from halobacterium-halobium-recording, measurement and evaluation of electronmicrographs at 3.5Å resolution," *Ultramicroscopy*, no. 19, pp. 147–178, 1986.
- [7] J. A. Mindell and N. Grigorieff, "Accurate determination of local defocus and specimen tilt in electron microscopy," *J. Struct. Biol.*, no. 142, pp. 334–347, 2003.
- [8] C. Toyoshima and N. Unwin, "Three-dimensional structure of the acetylcholine receptor by cryoelectron microscopy and helical image reconstruction," *J. Cell Biol.*, no. 111, pp. 2623–2635, 1990.
- [9] Z. Metlagel, Y. S. Kikkawa, and M. Kikkawa, "Ruby-helix: An implementation of helical image processing based on object-oriented scripting language," *J. Struct. Biol.*, 2006 (in press).

Public Interactive Display Using Front-projection And Infrared-pass Filter Camera

Cheng-Tse Chu, Dandi Duan and Richard Green

Department of Computer Science and Software Engineering, University of Canterbury

Email: {ctc20, ddd12}@student.canterbury.ac.nz, richard.green@canterbury.ac.nz

Abstract

Projectors have been traditionally used for making fixed displays, but with the advances in projector-camera technology, they can be integrated to allow transformation of any surface into a display screen, leading to increased opportunities for interactive ubiquitous displays. In this paper, we describe our implementation of using infrared-pass filter camera in a front-projection environment and how people in a public area can easily interact with the projected animation. The computer-vision based motion tracking techniques are based on background subtraction and double difference algorithm. Experience with using these techniques, the result of a user test, some design trade-offs and lessons, and future directions are discussed.

Keywords: Public interactive display, infrared-pass filter camera, front-projection

1 Introduction

Progressively, the advent of innovative sensing and display technology has made large interactive displays ubiquitous. They are found in shopping centres, railway stations and airports. Technology has encouraged the development of interactive displays to serve different purposes such as exchanging data, publishing information, and advertisement.

Computer vision can provide the basis for direct interaction because of its flexibilities. Since the complexity of general vision tasks has often been a barrier to widespread use in real-time applications, we simplify the task by using an architecture based on only background subtraction and double difference algorithm.

In this paper, we present a front-projected computer-vision based interactive floor system, which allows people to play with virtual objects projected on the ground. The system is set up with both projector and infrared camera in front of the projection surface. The aim of this project is to model better interaction by using different computer vision approaches to detect different user motions. For example, if a person approaches a ball swiftly, our program should detect that as a strong kick and model the ball motion to fast acceleration.

2 Configuration

There are three most common approaches [4] to set up an interactive display area (figure 1).

One is the top-down approach, where a camera and projector is mounted high on a shelf or ceiling. Such approach has the following drawbacks.

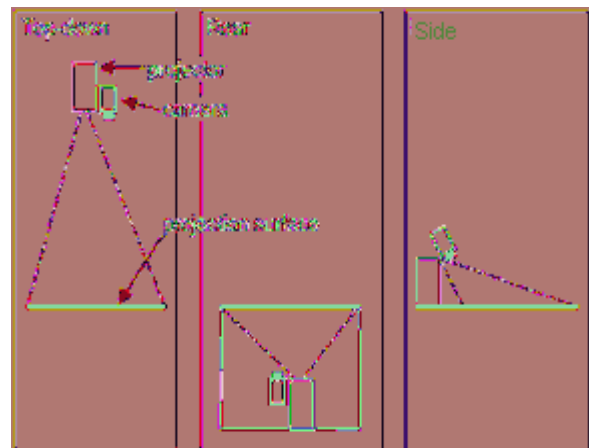


Figure 1. Three most common approaches for setting up vision-based projection systems. Projector and camera mounted from above (left), Rear projection with both camera and projector behind the projection surface (middle), and camera and projector mounted at the side of the active surface (right) [4].

- Heavy projector and camera are difficult to install on the ceiling and it requires special mounting hardware and is best left to professionals.
- The system and the projection surface have limited mobility.
- Minor vibrations can create problems and is difficult to maintain calibration.
- Since both the projector and camera are above the displaying area, the user's own body can occlude the projected image.

Another approach is rear projection. Both the projector and camera are placed behind an opaque projection surface. This effectively removes occlusion problems but this approach also has some drawbacks.

- The camera detects user motion through an opaque displaying surface; therefore the image resolution is limited.
- Such approach is not suitable for table-top interactive display because the result space for projector and camera can be quite large and difficult to fit under a table.
- A dedicated purpose opaque surface is required.

The third approach is suggested in PlayAnywhere. It is to mount both camera and projector to the side of the active surface. It can be set up and moved easily, but it has two potential drawbacks.

- Since the image is projected from the side to the projection surface, this can result in more image occlusions than top-down approach.
- The camera is looking at the projection surface from the side, so lens distortion is more obvious; therefore such an approach will require additional image processing.

Due to the expense of rear approaches and undesirable occlusion and distortion problems in side approach, our project uses the top-down approach. The projector is placed above the displaying area and images are projected to a mirror and reflected downward to the ground. This is because the heat may damage the device if the projector is facing downwards. The hardware setup of the projector is shown in Figure 2.



Figure 2. The projection reflects off a mirror onto the ground. Courtesy of LDPS NZ Ltd.

Our camera uses an infra-red (IR) low-pass filter to allow only infrared light pass through. Its purpose is to exclude projected animation and let the camera see only human motions. IR illumination is also used to illuminate the scene. A circular continuous density filter is applied to the IR light source to eliminate hotspots and obtain a more uniform illumination of the area.

For best performance, the animation is projected onto a flat surface vertical to the projection.

3 Related Work

There have been a great variety of studies on interactive tables, walls and floors [1]. One of the most famous interactive displays is LiveBoard, Tivoli. The purpose of LiveBoard [2] is to support group meetings, presentations and remote collaboration.

Information can be read regardless of the viewing angle, and a three-button mouse like pen is used for interaction. Tivoli is an application program which is implemented using Liveboard. It can be used like a whiteboard; additionally, the information can be saved, retrieved, printed and put on multiple pages.

Our project was inspired by HoloWall and Play Anywhere. HoloWall [3] is a large interactive display which allows people to exchange information in group meetings. The use of infrared light and a video camera can recognize body movement and trigger interactions.

PlayAnywhere [4] is a front projection interactive display. It has a number of contributions to image-processing techniques for front-projected vision-based table system; including a shadow-based touch detecting algorithm.

Ubwall [5] is another similar project; it is a large display system for advertisement and director services in public space. It is equipped with an RFID reader and an infrared motion sensor. Ubwall is adaptive that a user can put a RFID card on the reader, and then detailed and personalized information will be displayed.

VIDEOPLACE [6] is an artistic installation using a video camera that lets a user to interact with the environment using his/her body. ALIVE is also a vision-based interactive environments. A user can manipulate virtual objects by means of their own silhouette, so the interaction is indirect as compared to the HoloWall.

4 Video Processing

There are mainly three steps to achieve how the video taken by infrared camera is processed to interact with animation. Firstly, the camera needs to detect human and their motions within the projected area. Secondly, the matrices of camera frames that record human body shapes and motions need to be mapped with the projection. Finally, animated objects need to move or change in a way that reacts according to human motions.

4.1 Human Detection

Here we test two alternative foreground segmentation models. One is only subtracting human shapes from background image. The other one is further detecting human motions by double difference subtraction. The optimal solution depends on content and style of the animation.

4.1.1 Background Subtraction

To obtain human shapes from the scene, background subtraction is implemented. The default model averages the first few frames as a background image. This background image can be updated at any time by saving a new image. Since the camera filters all but infrared lights, it is set up so that no thermal objects are within the camera's view while such an image is acquired.

When a player steps into the projected area, he is shown inside the camera's angle of view. A thermal video of the player is taken by the camera. Each frame of the video is compared to background image using pixel-to-pixel subtraction, and the results are recorded into a matrix represented by a two dimensional array. This matrix does not only indicate if a particular pixel has changed from background image, but also keeps track of how adjacent pixels have changed. By counting up number of consecutively changed pixels and recording the value into the matrix, the camera is able to detect how fast the thermal object is moving and in what direction. The animated object can later react to the player according to the speed at which the person approaches.

4.1.2 Double Difference Subtraction

Another time differential algorithm, double difference, is used to extract moving points from image sequences. In this method, object motions with respect to previous positions are computed based on the hypothesis that some object points overlap in two consecutive frames [7]. Instead of subtracting background from current image, we calculate the pixel difference between current frame and last frame as well as the pixel difference between last frame and second last frame. A logical AND is then applied to these two differences, and the result is stored into an image matrix for later use [7]. Equations (1), (2) and (3) show this process:

$$PD_1 = I_1(x, y) - I_2(x, y) \quad (1)$$

$$PD_2 = I_2(x, y) - I_3(x, y) \quad (2)$$

$$DD_1 = PD_1 \text{ AND } PD_2 \quad (3)$$

If the newly captured image is I_1 and the last two images are I_2 and I_3 , for a particular pixel (x, y) , double difference DD_1 is obtained by computing two pixel differences PD_1 and PD_2 and then computing the conjunction of these results.

Comparing these two methods, we find that they serve well for different animations. If the interaction is between a player and a moving object (top image in figure 3), such as a ball, background subtraction is suitable to direct the object. For example, in a soccer game, when the player stops kicking and stands still, the ball should not move into the area under his feet. This is manipulated by background difference, as long as the player is inside the camera's view, his shape is

recognized and the ball is always outside of it.

Since double difference subtraction extracts moving points from image sequences, it is more suited to calculating motion vectors rather than static location. For example, if the animation is designed to allow user to play with water (bottom image in figure 3), once the user stops moving, all ripples should disappear slowly and the water should be still everywhere.

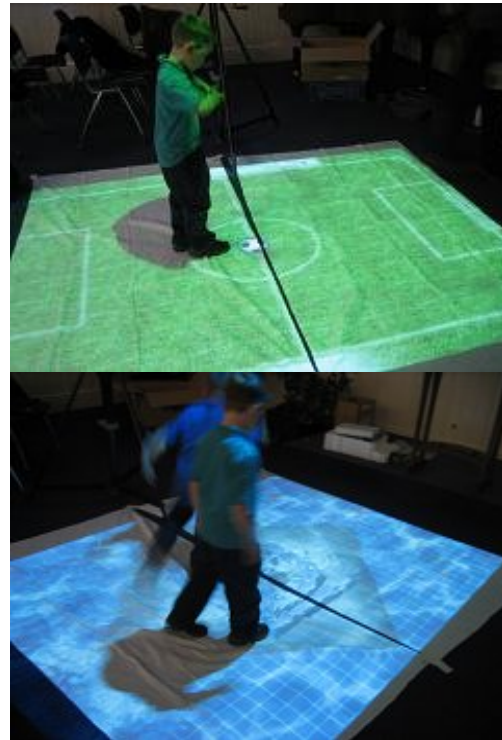


Figure 3. Top: a player is playing with virtual moving object. Bottom: players are playing with virtual water. Courtesy of LDPS NZ Ltd.

The other useful feature of double difference subtraction is that it is more robust to noise due to small camera movements. Since the camera may be hanging on the ceiling to get a top-down view, its angle of coverage may slightly change at times due to vibrations of the building. While using background subtraction, most of the time the background image is fixed. It may not be possible to acquire a background image only whenever it changes¹. If double difference algorithm is used instead, there is no permanent background image since motion detection involves with only most recent frames.

4.2 Mappings between Video and Animation

After human or their motions have been detected from camera's angle of view, which is same as the

¹ Although we are using infrared camera, the background image may slightly change over time due to camera movement or change of thermal devices, such as lighting and heating systems.

projected area, next step is mapping between video frame sequences and animation. Here we map the animation onto images.

Corners of each animated object are located. Their values of x-axis and y-axis at any certain time are translated into the image captured at the same time by multiplying x-axis ratio and y-axis ratio respectively. The translated results are then used to construct an area of the object inside camera's view, so overlaps between this area and detected human motions can be checked later. For example, if we have a rectangle animated object and the positions of its four corners are represented as (MinX, MinY), (MaxX, MinY), (MaxX, MaxY) and (MinX, MaxY), we then use equation (4), (5), (6) and (7) to calculate the object's position in camera image:

$$\text{CamMinX} = \text{MinX} * \text{RatioX} \quad (4)$$

$$\text{CamMaxX} = \text{MaxX} * \text{RatioX} \quad (5)$$

$$\text{CamMinY} = \text{MinY} * \text{RatioY} \quad (6)$$

$$\text{CamMaxY} = \text{MaxY} * \text{RatioY} \quad (7)$$

$$\text{RatioX} = \text{CamViewWidth} / \text{AnimationWidth} \quad (8)$$

$$\text{RatioY} = \text{CamViewLength} / \text{AnimationLength} \quad (9)$$

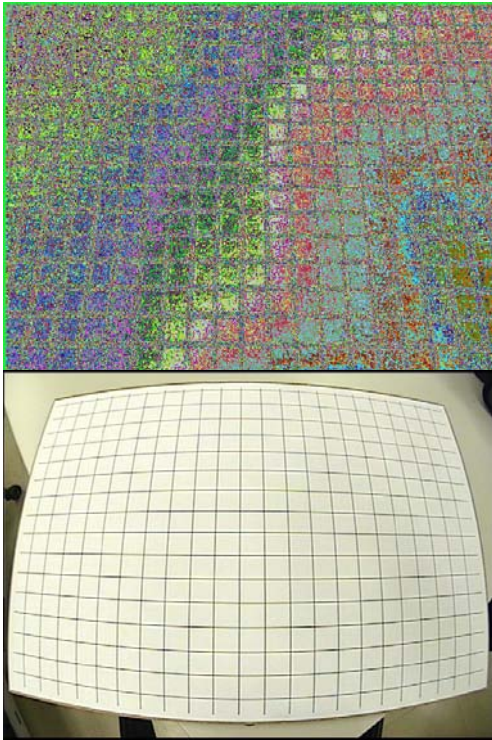


Figure 4. Top: the original image. Bottom: the outward distorted image. [10]

While mapping the animation onto images, there is a lens distortion problem. The image points, especially those close to edges of the image, are displaced outward from the optical center. Figure 4 compares an original image and a distorted image. This radical distortion is the most significant distortion, whose effects vary with distance from the optical center of

the lens [9]. To correct lens distortion, the model described in OpenCV documentation can be used [8], [9]:

$$x' = x/z \quad (10)$$

$$y' = y/z \quad (11)$$

$$x'' = x'(1+k_1r^2+k_2r^4)+2p_1x'y'+p_2(r^2+2x'^2) \quad (12)$$

$$y'' = y'(1+k_1r^2+k_2r^4)+2p_2x'y'+p_1(r^2+2y'^2) \quad (13)$$

$$r^2 = x'^2 + y'^2 \quad (14)$$

$$u = f_x x'' + c_x \quad (15)$$

$$v = f_y y'' + c_y \quad (16)$$

where x , y and z are coordinates of a 3D point in a world coordinate space, u and v are coordinates of a pixel in the image plane, f_x , f_y , c_x and c_y are calibration parameters, and k_1 , k_2 , p_1 and p_2 are the radial and tangential distortion coefficients.

4.3 Animation Updates

To move an animated object according to user's motion, or precisely speaking, to adapt the position of a displaying object, each pixel in its region has to be tested in order to modify movement parameters. If the recorded difference image overlaps with this region, x-axis movement parameter and y-axis movement parameter are cumulated respectively according to the direction of detected motion, that is, vertical or horizontal. A negative parameter indicates that the object is moving left or up, whereas a positive parameter causes the object to move right or down. This is decided by the position of where the overlap has happened.

Speed of movements is controlled by three values. The first value is that in each cell of the difference array where it was calculated accumulatively such that the deeper a detected shape or motion enters the object's region, the higher a value is. The movement parameter is multiplied by this value at each pixel so that differences between faster moves and slower moves can be demonstrated. The second value is a variable, called acceleration factor, which also modifies the moving animated object's velocity. It is used in a similar manner. By multiplying this factor, movement parameters can accelerate the object in a flexible way. The last value is a deceleration rate, which ensures that objects decelerate correctly when there is no more external force on it.

Note that all these animation movement parameters are set for only passive objects that do not move themselves, such as a soccer ball. For active objects, such as cartoon characters, they can move freely within the scene once they have been "touched" as long as they do not go inside where the player is standing.

The other issue is that whenever an animated object, either active or passive, reaches the edge of animation area, it does not go beyond that edge.

5 Result and Discussion

5.1 Result Evaluation

In order to evaluate the usability of the system, we designed a user study where we asked a number of participants to perform a simple task, which is moving a ball from the centre toward different directions in the field as shown in figure 5.

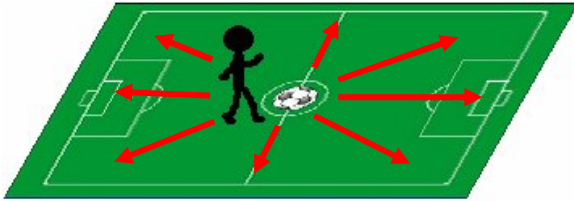


Figure 5. A simulation of usability test.

Two experiments have been carried out with different ball radii. The radius to screen width ratio was 1/32 in the first test and 1/16 in the second. For the first test, the accelerate factor was 0.5 and the deceleration factor was 0.8. For the second test, 0.2 was used for acceleration and 0.7 for deceleration. The goal of this experiment was to find out the relationship between virtual object size and its interaction performance.

Figure 6 shows the number of successful attempts in each direction with different radii. The figure suggests that radius size is proportional to detection accuracy. The larger the size of radius, the better the detection is. Program with the larger radius provides better detection in most directions.

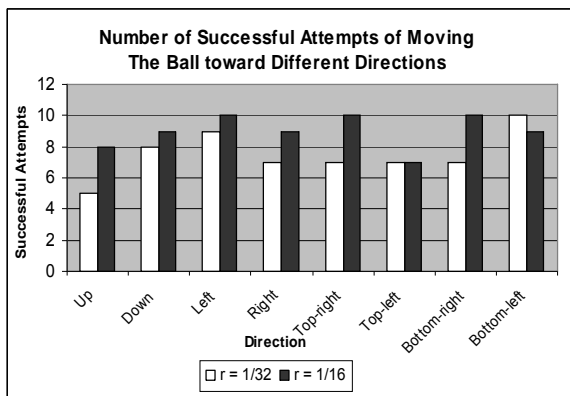


Figure 6. Number of successful attempts of moving the ball toward different directions

Comparing to previous works, which use only one detection algorithm, our system can target wider range of applications because we have two detection methods: background subtraction and double difference subtraction. As mentioned in section 4.1.2, they serve well for different type of motions.

We have noticed that most of the prior researches use at least two cameras for detection: one camera takes infrared images, and the other takes normal images of the real world [11], [12]. In Contrast, our system used only one infrared camera. It may be less robust for

detecting the exact contact of real and virtual objects because infrared images do not always provide a clear contour of human body. However detection in our system maybe faster because we use simpler detection approaches in which only one input camera is taken into account.

In our approach, the animated object is always approximated by a rectangular bounding box. The drawback is that there will always be a distance inaccuracy between the player and the animated object under the situation shown in Figure 7. We have carried out an experiment on the interaction between a soccer ball and a detected object.

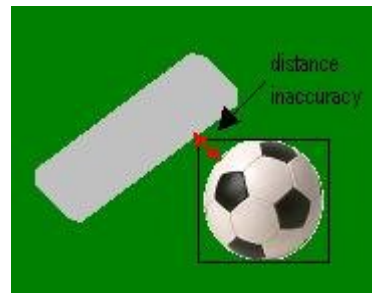


Figure 7. Distance inaccuracy of our approach.

From the experiment, we found the maximum distance inaccuracy occur when the moving object approached the soccer ball diagonally and coincided with the bounding box. When setting the ball radius to twenty units, the maximum distance inaccuracy MDI is about 8.2843 units. The calculation of MDI is shown in equation (17):

$$MDI = \sqrt{2r^2} - r \quad (17)$$

5.2 Pros and Cons

The use of infrared-pass filter camera allows separation of animations and human motions. As a result, animations do not affect the performance of computer-vision based tracking. Unlike rear projection systems, our system does not require dedicated surfaces for projection; animation can be projected on to any plain flat surface. Our computer vision based tracking is based on background subtraction models; therefore multiple objects can be detected at the same time. This suggests that it has the potential to cater for multiplayer requirements.

One potential problem of using an infrared-pass filter camera for tracking is that it is difficult to distinguish different body parts of a person. For example, if a user tries to kick a ball, but in camera's view, his head touched the ball first, the ball would be moved differently from user's intention. One drawback of infrared-filtered camera is that it can reduce contrast; it may not provide as sharp a silhouette edge as a normal camera without added filter. Therefore it may be difficult to detect the exact shape of the body in contact with animated objects and based on that, model correct movements.

6 Conclusion

Our system combines front-projection and infrared-pass filter camera to build an interactive display system. Users can interact with different animations by moving their body parts inside the camera's view. Background subtraction or double difference algorithm is then used to detect these users. Difference images are mapped with the display so that animated objects can react according to users' movements.

7 Future work

Depending on the angle of the lens, the video may have significant distortion caused by camera lens. By using image processing, the input image and projected image can be further aligned to improve tracking performance.

If the infrared camera is set in a place where thermal conditions change over time, various adaptive background subtraction techniques may be applied in order to guarantee the quality of background difference image [13].

8 Acknowledgement

We thank Ray Hidayat for prior work on interactive displays. We also thank our supervisor Dr. Richard Green and LDPS NZ Ltd for their valuable comments.

9 References

- [1] K. Meyer, Ambient Display and Their Interaction Techniques – Review and Synthesis, 2006.
- [2] Scott E, Richard B, Rich G, David G, Frank H, William J, David L, Kim M, Elin P, Ken Pier, John T, and Brent W. Liveboard: a large interactive display supporting group meetings, presentations, and remote collaboration. In *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 599–607, New York, NY, USA, 1992. ACM Press.
- [3] Nobuyuki M and Jun R. Holowall: designing a finger, hand, body, and object sensitive wall. In *UIST '97: Proceedings of the 10th annual ACM symposium on User interface software and technology*, pages 209–210, New York, NY, USA, 1997. ACM Press.
- [4] Andrew D.W, PlayAnywhere: A Compact Interactive Tabletop Projection-Vision System, 2005.
- [5] Minoru S, Hirohisa N, Akinobu U, Toru O, and Masao Y. "ubwall", ubiquitous wall changes an ordinary wall into the smart ambience. In *sOc-EUSAI '05: Proceedings of the 2005 joint conference on Smart objects and ambient intelligence*, pages 47–50, New York, NY, USA, 2005. ACM Press.
- [6] Mark A, Laurie M, Masood M, Lance P, Malcolm P, Bill R and Kirsten T. Use of Video Shadow for small Group Interaction Awareness on a Large Interactive Display Surface, 2002.
- [7] R. Cucchiara, M. Piccardi, A. Prati, N. Scarabottolo. Real-Time Detection of Moving Vehicles, 1999.
- [8] Benjamin A. Ahlborn, David Thompson, Oliver Kreylos, Bernd Hamann, and Oliver G. Staadt. A Practical System for Laser Pointer Interaction on Tiled Displays. *Proceedings of ACM Virtual Reality Software and Technology*, 2005
- [9] Gadi Glogowski, Johann Sawatzky. Computer Vision Measurements of Brake Shoes for Fort Garry Industries, 2006.
- [10] Toru Tamaki, Tsuyoshi Yamamura, and Noboru Ohnishi. Correcting Distortion of Image by Image Registration. *Proceedings of ACCV2002: The 5th Asian Conference on Computer Vision*, Vol. II, pages 521-526, Jan. 2002.
- [11] Lijun Jiang, Feng Tian, Lim Ee Shen, Shiqian Wu, Susu Yao, Zhongkang Lu, and Lijun Xu. Perceptual-based fusion of ir and visual images for human detection. *International Symposium on Intelligent Multimedia, Video and Speech Processing*, pages 514– 517, 2004.
- [12] Neeti A. Ogale. A survey of techniques for human detection from video, 2005.
- [13] Alan M. McIvor. Background Subtraction Techniques. *Proceedings of IVCNZ00*, 2000.

Simulation of multi-polarisation SAR imagery

S.J. McNeill, D. Pairman, H.C. North, S.E. Belliss

Landcare Research, Box 40, Lincoln 7640, New Zealand.

Email: mcneills@landcareresearch.co.nz

Abstract

We propose a SAR image simulating method developed to help SAR image analysis and training. The main features of the simulation method are the capability of wide area simulation and relatively low computational costs. The cost efficiency is achieved by using a hybrid method that assigns radar cross section values thematically where it is necessary to do so. However, the method allows for more detailed modelling of classes if possible. In this case, a regression model has been used to estimate the underlying radar cross for the exotic forest class, as a function of forest stand species and stand age. Speckle is simulated by a multivariate gamma distribution, with multi-variate parameters estimated from reference AIRSAR imagery. Finally, the radar noise floor is simulated by an additive Gaussian process.

Keywords: SAR, radar, simulation, multi-polarisation

1 Introduction

Synthetic aperture radar (SAR) is useful for many applications in environmental remote sensing, including topographic mapping, and land cover mapping. Multi-polarisation SAR is particularly useful since this form of data provides additional degrees of freedom for use in a classification, or in a multi-variate regression model [1]. However, multi-polarisation SAR images require a great deal of experience for correct interpretation, since the appearance of these images is very different from multi-spectral optical imagery, such as that available from the Landsat or SPOT satellite series. Furthermore, the appearance of the multi-polarisation SAR image changes significantly with the observation conditions, such as the incidence angle, and the interaction between these observational conditions and the local topography.

Simulated SAR imagery is designed to bridge the gap between the future availability of real data, and a user's desire to understand what the data will look like before committing resources to a future data source. Simulation has a particular advantage in the case of space-borne SAR systems, since the time required to develop an application using such data sources can be long, perhaps measured in years.

If simulation is not used, users may need to wait for the launch of the SAR system to gain access to suitable test data, in which case valuable operational time may be lost. Simulation can also reduce the cost of an initial study, since many different imaging scenarios may be investigated, while the

purchase of real imagery for these different scenarios may be prohibitively expensive.

Practical SAR image simulation represents a compromise between the need to provide a product that is detailed enough to reproduce the basic functionality required by the user community, but not so detailed that the user is unlikely to ever use the intricate functionality that is included. Thus, the type of simulation that is undertaken is closely tied to the likely end-user application.

In the present case, the simulated imagery is required for an assessment of SAR imagery to separate mature forest stands from cleared or harvested forest stands. This problem arises in commercial forest inventory work, as well as carbon accounting activities in respect of international agreements, such the Kyoto agreement.

Previous research work by us has established that single-polarisation SAR imagery in the C- (5.6 cm) and L-band (24 cm) wavelengths is unable to separate mature from cleared forest stands, for the dominant species used in commercial forestry in New Zealand (*Pinus radiata*), while multi-polarisation SAR imagery at C- and L-band *can* be processed to separate mature from cleared forest stands, provided that certain processing steps are carried out [1]. The simulation effort described here is intended to allow users to become familiar with the various products that are expected to be available from future operational SAR systems, and which could be used in forest clear-cut work. The availability of this simulated imagery is likely to direct future research effort.

2 Simulation methodology

There are many examples of simulation methodologies for multi-polarisation SAR imagery in the literature. In essence, however, the available methods fall between the extremes defined by two categories. First, those methods that model the detected radar cross section (RCS) from a detailed description of the elements of the target and the interactions between those elements at a very detailed level (the RCS *forward prediction* method). Second, those methods that model the RCS using a categorical description of the target, and a simple assignment of RCS based on the target category (the RCS *thematic assignment* method).

Forward prediction methods [2, 3, 4] provide a comprehensive account of the target RCS, but the models can be computationally demanding, and very difficult to parametrise. Thematic assignment models generally produce RCS estimates derived from a land-cover classified image, along with known or published accounts of the RCS and polarisation-ratio for different vegetation types [5, 6]. A variant of this approach is to acquire a single-polarisation SAR image of one polarisation close in time to the land-cover classification, and derive the RCS for other polarisations from published accounts [7] of the ratio of RCS for different polarisations. For example, Buckley [8] has simulated Radarsat-2 imagery by using a Landsat-Thematic-Mapper-derived classification to provide a thematic base, as well as polarisation ratios tabulated in the literature for vegetation types similar to those found in the area, and a Radarsat-1 image of the area collected almost simultaneously with one of the Landsat images.

Thematic assignment models are considerably easier to implement than forward prediction models, and may be most suitable if the relationship between the RCS from different polarisations is described simply, such as by a ratio or by a shift in level. However, if the polarisation interaction is more complicated, then this simple methodology is unlikely to produce a fair representation of the target RCS behaviour. However, there is a strong incentive to keep the model as simple as possible, since it is very easy to produce a model that is computationally intractable, or difficult to parametrise.

In this paper, the important targets are shrubland and forest, and it can be shown that there is a strong interaction between the RCS of the different polarisations [1]. However, the correlation is not perfect (i.e the multivariate RCS is not degenerate), and there is a differential polarisation response with stand age that can be exploited to es-

timate biophysical parameters associated with the forest [1]. Thus, correct simulation of this type of target requires parametric information, in addition to the thematic class to which the target is assumed to belong.

For this present simulation effort, the test area is large (on the order of millions of hectares), so there is little hope of being able to model the detailed RCS relationships for every pixel in every thematic class. A more practical approach is to use the RCS thematic assignment method where the class is less relevant to the end-user application, but adopt a more rigorous method of RCS modelling in areas of production forest. Although the RCS modelling in these forest areas is not as rigorous as those used in the RCS forward prediction models, the complexity is sufficient to reproduce the main characteristics of the interactions that occur as the forest stand ages.

The basic idea of multi-polarisation SAR simulation described here is that the image space is partitioned by a non-overlapping spatial classification of targets. A model is then defined for each unique target class, and the parameters for each model are defined on a pixel-by-pixel basis. The class model, and its associated parameters, define the underlying target RCS for each required polarisation. For some classes, no parameters are required, either because they are not available, are not important to the end-user application, or are simple enough to be assigned simple values. For example, it is reasonable to assume that water bodies and smooth road surfaces should have a zero-value RCS. By contrast, grasslands may be assigned a nominal RCS value, since detailed parametric characterisation of the RCS is not available.

An important consideration in respect of multi-polarisation imagery is the fluctuation associated with the coherent SAR illumination (speckle), which is combined with the underlying RCS to form the measured RCS. For the number of looks usually employed with commercial SAR imagery (1–4 looks), speckle can usefully be modelled as gamma-distributed for each polarisation RCS, but the joint distribution is more difficult to describe in practice, since it depends strongly on the target, especially for those targets with significant spatial texture (e.g forest). In the SAR simulation method described here, these interactions are simplified by generating the speckle as a multivariate gamma distribution [10] with the variance-covariance matrix either measured from reference multi-polarisation imagery, assigned from tabulated values from the literature, or assigned by theoretical considerations (as appropriate for the class).

In addition to its point-by-point statistics, an important feature of multi-polarisation SAR is its spatial autocorrelation function (ACF), which represents spatial variation in targets such as forest, urban areas, or ocean waves [11]. Here, an important restriction on the simulated radar imagery is that this spatial structure is *not* simulated. This is partly due to complexity, since simulation of spatial ACFs in forest targets are particularly complicated. However, it is also argued that in the forestry applications for which this simulated imagery is intended, whole-stand classifications between mature and clear-cut stands are desired, rather than a detailed pixel-by-pixel classification. This simplification of the clear-cut stand application is outlined in somewhat more detail elsewhere [1].

Finally, it is noted that the observed RCS is always corrupted by noise, primarily additive receiver noise, and it is useful to have a means of quantitative comparison between this noise and the desired signal. For SAR, the additive noise component is expressed as noise-equivalent radar cross section (or noise-equivalent sigma-nought) $NE\sigma^0$. Image targets that fall below the $NE\sigma^0$ are not useful, and this level is simulated as an additive random Gaussian process. The point-by-point simulation method is shown in figure 1, producing the statistical estimate of the RCS in ground-range form.

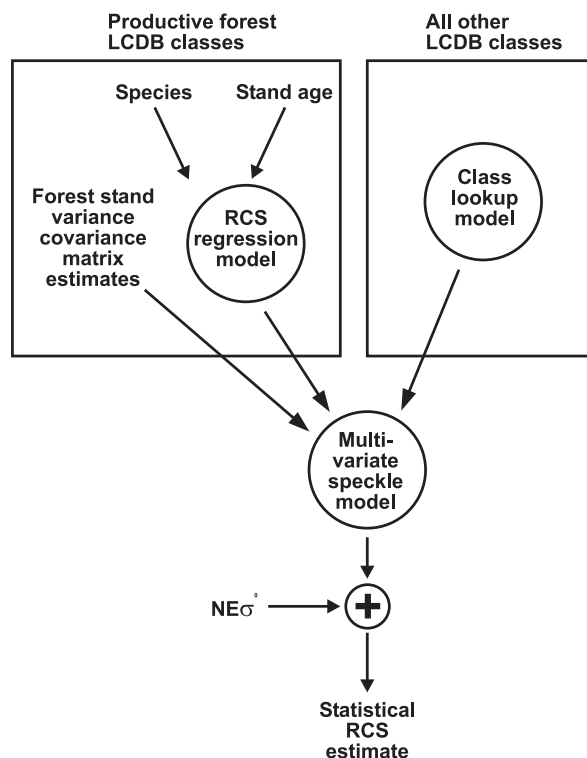


Figure 1: Schematic overview of the multi-polarisation SAR simulation methodology.

3 Simulation example

In this section, we describe the simulation of two recent multi-polarisation SAR systems. The first, Radarsat-2, operates at the short wavelength of 5.6 cm, while the second operates at a medium wavelength of 23.5 cm. Both SAR systems are candidates for forest clear-cut mapping in New Zealand.

3.1 Target SAR systems

The Phased Array L-band (23.5 cm wavelength) Synthetic Aperture Radar (PALSAR) is a SAR sensor on the Japanese Advanced Land Observation Satellite (ALOS), launched in January 2006. ALOS-PALSAR [12] is a considerably-enhanced version of an earlier Japanese satellite JERS-1, which operated from late-1992 to October 1998. This new sensor provides multi-polarisation imagery, a range of target viewing angles, several different imaging modes, and enhanced radiometric performance.

Radarsat-2 [13] is the follow-on mission to Radarsat-1, and at the time of writing is due to be launched from Russia's Baikonur Cosmodrome in Kazakhstan in March 2007. Radarsat-2 is a C-band (5.6 cm wavelength) SAR sensor that will be able to image at spatial resolutions ranging from 3 to 100 metres with nominal swath widths ranging from 10 to 500 kilometres. In addition, Radarsat-2 will offer multi-polarization imagery, a capability that helps in classifications of a wide variety of targets.

Although both ALOS-PALSAR and Radarsat-2 offer single-, dual- and multi-polarisation modes of interest to many users, in terms of this paper, the multi-polarisation mode is of most interest. The relevant multi-polarisation characteristics of ALOS-PALSAR and Radarsat-2 are shown in table 1, for indicative scene conditions.

3.2 Scene information

The test area is located in Kaingaroa Forest, centred on latitude $-38^{\circ}31'$, longitude $176^{\circ}32'$. Kaingaroa is New Zealand's largest production forest and is the second largest planted forest in the world, situated on the Volcanic Plateau in New Zealand's central North Island. This area has imagery available from the NASA AIRSAR [14] multi-wavelength, multi-polarisation aircraft mission flown in 2000 (PACRIM-II) [15]. NASA JPL supplied the AIRSAR imagery in compressed Stokes matrix form [16], with nominal intensity and phase calibration applied. Scenes are a multiple of 10 km in length, and approximately

Table 1: Indicative characteristics of the multi-polarisation modes of ALOS-PALSAR and Radarsat-2.

Parameter	ALOS-PALSAR	Radarsat-2
Product level description	Level 4.1, standard quad-pol.	Standard quad-pol.
Frequency (wavelength)	1.275 GHz (23.5 cm)	5.405 GHz (5.55 cm)
Noise-equivalent sigma-nought $NE\sigma^0$	≈ -23 dB	-31 ± 2 dB
Number of looks (range \times azimuth)	1×4	1×4
Ground sampling (cross \times along-track)	30×30 m	25×28 m
Ground scene size (cross \times along-track)	30×70 km	$25\text{--}50 \times 25$ km
Available local incidence angle	8.9–33.7 degrees	20–41 degrees

11 km in width, with a ground resolution of 10 m. The AIRSAR imagery has relatively high spatial resolution, a relatively high number of effective looks, and a lower $NE\sigma^0$ when compared to the equivalent parameters for either ALOS-PALSAR or Radarsat-2. Therefore, we expect that simulated versions of these future spaceborne missions represent degraded versions of the AIRSAR imagery.

Thematic information on the test area is provided by the Land Cover Data Base 2 (LCDB-2). LCDB-2 is derived, in part, from satellite imagery acquired in 2000/2001, and is a hierarchical development of the classes used for LCDB-1 (imagery acquired in 1996/1997) [17].

Parametric information within the test area was generated from an ArcInfo GIS coverage provided by Fletcher Challenge Forests, then owners of Kaingaroa Forest. This GIS coverage provided a number of attributes for each forest stand, but for the purposes of this study, only two parameters were used. The first was the forest species, Radiata pine (*Pinus radiata*) or Douglas fir (*Pseudotsuga menziesii*), while the second was the stand age in years at the time of the AIRSAR imagery. Areas outside the AIRSAR coverage region had no parametric information associated with them. The GIS coverage was converted to raster form, thus providing a pixel-by-pixel parametric description of the productive forest class in the test area.

4 Results

Figure 2 shows the LCDB cover classes, forest stand age, forest species, and AIRSAR L-band quad-polarisation SAR components for the study area. The GIS coverage contained forest stands from clear-cut to age 70 years, although most stands are less than 35 years old. For the LCDB, a total of 22 separate classes were obtained, although the test area is dominated by exotic forest, with much smaller areas of pasture, indigenous forest, and bare surfaces (towns, roads etc). Several of the LCDB cover classes were aggregated to a

single class, such as the various water bodies, and less-relevant distinctions between grasslands. For all but the forest classes, the RCS was defined by estimates provided from the literature, or from other AIRSAR images. Some classes (e.g water, roads) were assigned a zero-valued RCS, since that is the most plausible figure.

As noted earlier, the regressions for RCS in the forest class were defined in terms of the species and the forest stand age. The regression analysis that defined these relationships was defined from a sample of 465 stands from Kaingaroa forest, with 403 stands in Radiata pine and 62 in Douglas fir with an area over 10 ha. A summary of the regression analysis for C- and L-band is given in table 2. The important result from these results is that the RCS (dB) for a given polarisation can be estimated with a standard error of approximately 2 dB, using a linear relationship between $\log(\text{Stand age})$ and an offset allowance for the difference in species. The correlation estimates for C- and L-band are roughly comparable, except that for L-HH, which is quite poor. The coefficients for the intercept and $\log(\text{Stand age})$ were highly significant. The coefficient for the effect of the species was highly significant, except for L-HH and L-VV.

Finally, the noise-floor of the simulated radar was estimated by using the mean expected value of $NE\sigma^0$ from table 1. The final RCS statistical estimates were converted to log-compressed σ^0 values in dB, as is standard in the literature.

5 Discussion

It is difficult to assess the quality of the final simulated result, for several reasons. First, the RCS estimates are designed to be consistent with, but not identical to the values from the AIRSAR imagery. This consistency is defined by the statistics of the regression model used to relate stand age and species to RCS value, in the case of the production forestry class. The actual values of RCS in the AIRSAR stand will differ by some random amount that will vary from stand to stand. However, the

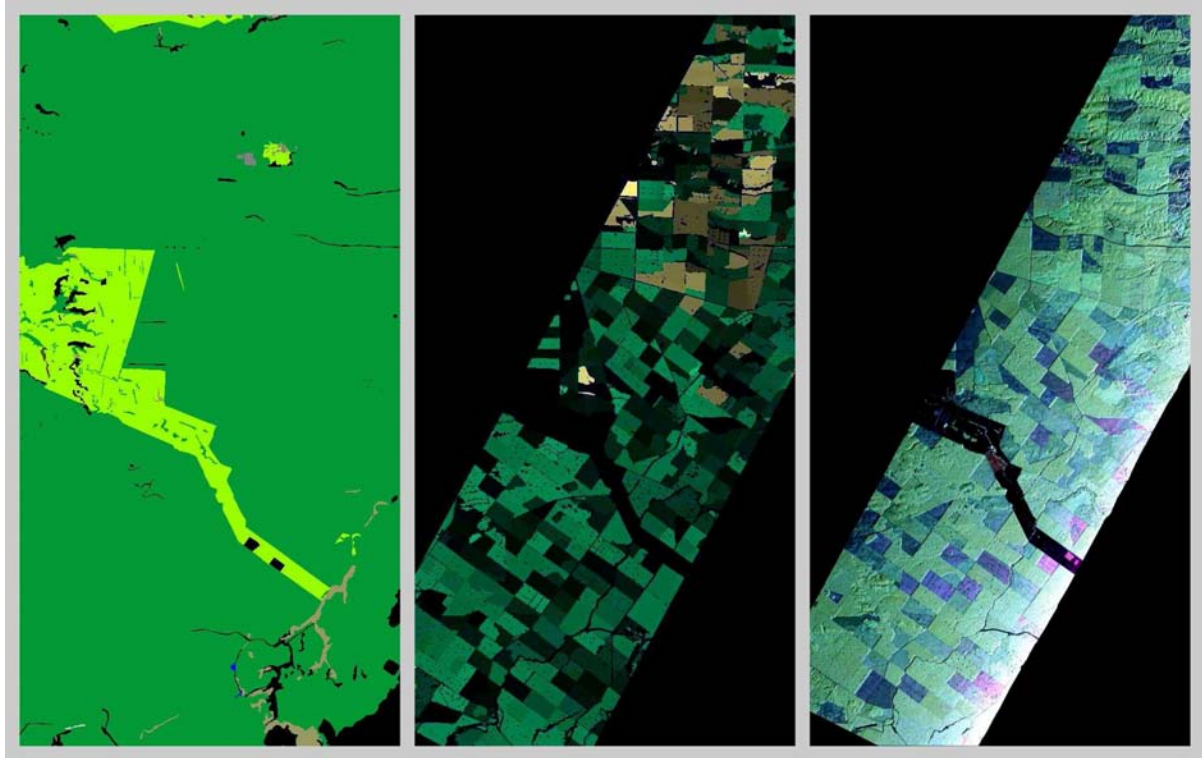


Figure 2: Simulation study area, Kaingaroa forest, North Island, New Zealand. Left: LCDB cover classes, colour-coded. Centre: Forest species (Green – Radiata pine, Brown – Douglas fir) with forest stand age coded by brightness, with maximum stand age 70 years. Right: AIRSAR L-band quad-polarisation SAR image (RGB – HH, HV, VV).

overall difference between the AIRSAR image and the simulated values is expected to be unbiased.

Second, the RCS assignment is made on the assumption that there is a one-to-one relationship between the LCDB class and the type of scattering that occurs within that class. However, this is unlikely to be the case. For example, in towns, the type of scattering will vary considerably between different areas of the town, changing from Rayleigh scattering to double-bounce scattering, depending on the nature of the scene. There is little that can be done to solve this problem, since finer thematic classifications are not available.

Similarly, for forest stands it is possible that the modelling used here is too simplistic to account for changes evident in different stands. Possible effects that may be important, but which have not been considered are: stand density (stems per unit area), whether pruning had been involved, and whether other forest treatment factors have been used. Aside from these above cautions, the RCS estimates are in line with the broad range of values predicted by the RCS regression procedure.

6 Conclusions

The simulation method described here is a hybrid method that uses the simplicity of the RCS thematic assignment method where it is required, but allows for more detailed modelling if possible. A regression model of RCS has been used for the exotic forest class, which is the important class for this study, and this model is based on parameters of stand species and stand age. Speckle is simulated by a multivariate gamma distribution, with multi-variate parameters estimated from reference AIRSAR imagery. Finally, the radar noise floor is simulated by an additive Gaussian process.

7 Acknowledgements

This research was funded by the Foundation for Research, Science and Technology.

We acknowledge NASA Headquarters, NASA Jet Propulsion Laboratory, and NASA Dryden for provision of administration, instrumentation and processing, and flying facilities, respectively, during the 2000 PACRIM-II mission over New Zealand.

Table 2: Summary of regression results for C- and L-band backscatter as a function of forest parameters.

Band	Polarisation	R^2	p-value			SE (dB)
			Intercept	Log(Age)	Species	
L	HH	0.100	$< 2e - 16$	$7.87E - 12$	0.52	2.1
L	HV	0.591	$< 2e - 16$	$< 2e - 16$	0.00248	2.0
L	VV	0.403	$< 2e - 16$	$< 2e - 16$	0.962	1.9
C	HH	0.386	$< 2e - 16$	$< 2e - 16$	$< 2e - 16$	2.1
C	HV	0.528	$< 2e - 16$	$8.37E - 13$	$< 2e - 16$	1.4
C	VV	0.356	$< 2e - 16$	$2.31E - 12$	$< 2e - 16$	1.9

References

- [1] S. McNeill and D. Pairman, "Stand age retrieval in production forest stands in new zealand using C- and L-band polarimetric radar," *IEEE Transactions on Geoscience & Remote Sensing*, vol. 43, no. 11, pp. 2503–2515, 2005.
- [2] F. Ulaby, K. Sarabandi, K. McDonald, M. Whitt, and M. Dobson, "Michigan microwave canopy scattering model," *International Journal of Remote Sensing*, vol. 11, no. 7, pp. 1223–1253, 1990.
- [3] P. Stuopis, J. Henson, R. Davis, and K. Hall, "Modelling of forested area for real and synthetic aperture imaging radar simulation," in *Proc. IGARSS-1996*, vol. 1, (Lincoln, Nebraska, USA), pp. 254–256, July 1996.
- [4] I. Koh and K. Sarabandi, "A new approximate solution for scattering by thin dielectric disks of arbitrary size and shape," *IEEE Transactions on Antennas and Propagation*, vol. 53, no. 6, pp. 1920–1926, 2005.
- [5] M. Tsuchida, K. Suwi, and M. Iwamoto, "A simulator of synthetic aperture radar images; land, ocean surface, and man-made targets," in *Proc. IGARSS-2004*, vol. 7, (Anchorage, Alaska, USA), pp. 4836–4838, Sept. 2004.
- [6] Y. Yang, H. Ewe, C. Hsu, L. Wang, S. Shih, K. Ding, J. Kong, R. Shin, H. Nguyen, T. Nguyen, J. Ho, and K. O'Neill, "A GIS-driven interactive radar image simulation using EMSARS model," in *Proc. IGARSS-1994*, vol. 2, (Pasadena, California, USA), pp. 857–859, Aug. 1994.
- [7] F. Ulaby and M. Dobson, *Handbook of radar scattering statistics for terrain*. Norwood, Mass.: Artech House, 1989.
- [8] J. Buckley, "Enhanced classification of prairie landscapes using simulated Radarsat-2 imagery," *Canadian Journal of Remote Sensing*, vol. 30, no. 3, pp. 510–516, 2004.
- [9] G. Ronning, "A simple scheme for generating multivariate gamma distributions with non-negative covariance matrix," *Technometrics*, vol. 19, no. 2, pp. 179–183, 1977.
- [10] D. Blacknell, A. Blake, O. Lombardo, and C. Oliver, "A comparison of simulation techniques for correlated Gamma and K-distributed images for SAR applications," in *Proc. IGARSS-1994*, vol. 4, (Pasadena, California, USA), pp. 2182–2184, Aug. 1994.
- [11] H. Wakabayashi, N. Ito, and T. Hamazaki, "PALSAR system on the ALOS," in *Proc. SPIE Sensors, Systems, and Next-Generation Satellites II*, vol. 3498, (Barcelona, Spain), pp. 181–189, Dec. 1998.
- [12] L. Morena, K. James, and J. Beck, "An introduction to the RADARSAT-2 mission," *Canadian Journal of Remote Sensing*, vol. 30, no. 3, p. 221234, 2004.
- [13] Y. Lou, T. L. Akins, D. A. Imel, T. W. Miller, D. Moller, *et al.*, "Review of the NASA/JPL airborne synthetic aperture radar system," in *Proc. IEEE Int. Geosci. Remote Sensing Symp. (IGARSS'02)*, (Toronto, Canada), pp. 1702–1704, June 2002.
- [14] I. Tapley, A. Milne, and E. E. O'Leary, "An overview of the PACRIM 2000 airborne synthetic aperture radar (AIRSAR) mission in the Pacific, Australia and Asian region," in *Proc. IGARSS-2001*, vol. 3, (Sydney, Australia), pp. 1387–1388, July 2001.
- [15] P. C. Dubois and L. Norikane, "Data volume reduction for imaging radar polarimetry," in *Proc. IEEE Int. Geosci. Remote Sensing Symp. (IGARSS'87)*, (Ann Arbor, MI), pp. 691–696, 1987.
- [16] Ministry for the Environment, "The Land Cover Classes." <http://www.mfe.govt.nz/issues/land/land-cover-dbase/classes.html>, visited on 15-Sep-2006.

Extracting Surface Curvature from Noisy Scan Data

J. Rugis^{1,2}

¹CITR, Dept. of Computer Science, University of Auckland.

²Dept. Electrical & Computer Engineering, Manukau Institute of Technology.

Email: john.rugis@manukau.ac.nz

Abstract

In general, the noise that is present in real-world 3D surface scan data prevents accurate curvature calculation. In this paper we show how curvature can be extracted from noisy data by applying filtering after a noisy curvature calculation. To this end, we extend the standard Gaussian filter (as used in 2D image processing) by taking adjacent point distances along the scanned surface into account. A brief comparison is made between this new *2.5D Gaussian filter* and a standard 2D Gaussian filter using data from the Digital Michelangelo Project.

Keywords: Surface curvature, noisy scan data, 3D noise filtering

1 Introduction

Three dimensional objects are often digitized in a way that results in surface point data sets [1]. In general, real-world scan data is noisy due to inaccuracies accumulated in the scanning process¹. Curvature, being a second derivative property, is particularly sensitive to corruption by noise.

A common approach towards solving this problem is to smooth the point data prior to attempting curvature calculations. This approach has a shortcoming in that surface detail can easily be lost. Alternatively, in this paper, we firstly calculate *noisy curvatures* and subsequently apply filtering to these curvature values.

Although the standard 2D Gaussian filter can be used in our approach, it does have the undesirable effect of smoothing edges as well as noise. Edge preserving variants of the 2D Gaussian filter have been developed. The bilateral filter by Tomasi and Manduchi [2] is a 2D filter that employs an edge preserving term that decreases pixel weighting based on pixel intensity differences.

Smoothing of the 3D shape itself is the goal of other noise reduction filters. A curvature and Laplacian operator based diffusion approach was introduced by Desbrun et al. in [3]. Work by Fleishman et al. consists of a mesh de-noising algorithm that operates on a surface predictor geometric component of the mesh [4].

In contrast, even though the approach that is presented in this paper uses 3D surface point position

¹In this paper, we will consider noise which has primarily a Gaussian distribution.

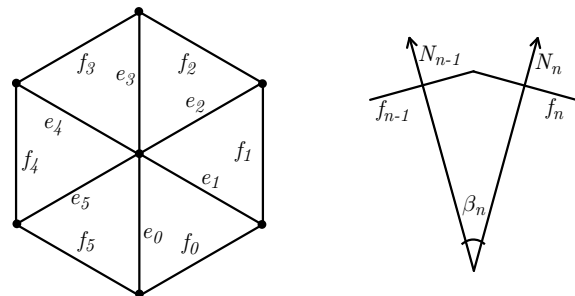


Figure 1: Curvature estimators: point adjacency on the left and face normals on the right.

information, it does not alter the position of those points.

2 Surface curvature estimators

Surface curvature is a well-defined property for continuous smooth surfaces [5]. However, when working with point data sets, many surface properties can only be estimated [6], and there exists a number of different estimators for determining curvature [7].

In this paper, the mean curvature is estimated as done by other authors [8]. With reference to the left side of Figure 1, we consider a point on the surface and, say, six adjacent points. The points are thought to be connected by *edges*, and edges enclose, in this case, six *faces*. We also identify an area $\mathcal{A}(f_n)$ associated with each face f_n . On the right side of Figure 1, we identify a surface normal vector associated with each face from an edge-on view point. The angle between adjacent face normals is designated as β . Angle β is positive if the faces form a convex surface (i.e., when viewed

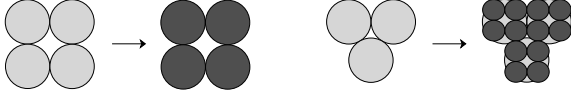


Figure 2: Mappings of orthogonal (on the left) or hexagonal (on the right) grids into an orthogonal grid.

from the outside) and β is negative if the faces form a concave surface (i.e., when viewed from the outside).

The mean curvature at the point P is estimated by

$$H(P) = \frac{3 \sum \|e_n\| \beta_n}{4 \sum \mathcal{A}(f_n)} \quad (1)$$

This estimator is generally valid, without change, in the case of adjacency point counts other than six.

3 Curvature maps

For 2D visualization purposes it is useful to convert the mean curvature values at surface scan points into a (2D) *curvature map* [9]. If the 3D point data has been acquired in a 3D orthogonal grid, then the curvature mapping is straightforward (defined by orthogonal cuts parallel to coordinate planes, see [10]). For data that has been acquired in a hexagonal grid, a *squashed dot* mapping is used [9].

Mappings for both cases, either orthogonal or hexagonal, are shown in Figure 2. The second mapping has the expense of quadrupling the number of pixels.

4 Curvature noise filtering

Consider sampling the planar surface, which has zero curvature everywhere. Noisy sampled data points will exhibit a symmetrical distribution of positive and negative curvatures centered around zero, with the limiting mean value for many points being zero. This suggests filtering that includes a mean calculation.

In this paper we describe and briefly compare two different weighted mean based curvature noise filtering approaches.

4.1 2D Gaussian filter

In this approach, we start by converting the noisy curvature values into a 2D curvature map as described briefly in Section 3. See [9] for further details on the curvature map creation process. This conversion into 2D enables the use of standard 2D images processing techniques.

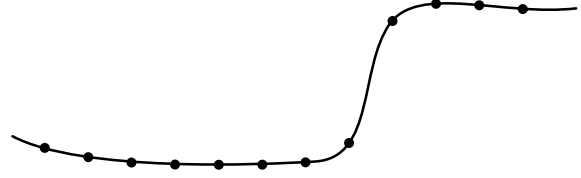


Figure 3: Cross section of a surface fold.

Next we apply a standard 2D image processing Gaussian filter which performs the desired smoothing based on the now fixed adjacency (and distance) relationships in the curvature map. The standard 2D Gaussian filter is implemented as a convolution process with the terms in an $(2m + 1) \times (2m + 1)$ Gaussian convolution kernel centered at $(0, 0)$ being determined using the formula

$$h(n_1, n_2) = h_g(n_1, n_2) / \sum_{n_1=-m}^m \sum_{n_2=-m}^m h_g$$

$$\text{with } h_g(n_1, n_2) = e^{-(n_1^2 + n_2^2)/2\sigma^2} \quad (2)$$

where, as usual, the standard deviation σ acts as a pixel area smoothing factor. Note that the $(n_1^2 + n_2^2)$ term can be thought of as the *adjacency distance* (squared) in a fixed adjacency grid.

4.2 2.5D Gaussian filter

As a preliminary motivation, note that the 2D filter, with its innate fixed distance adjacency, has the undesirable effect of smoothing edges which may be present due to silhouettes, occlusions, and surface folding in the scan. For example, Figure 3 shows a cross-section view of a surface fold in which the distances between adjacent scan points are not equal.

We introduce a *2.5D Gaussian filter* which gives consideration to edges. We will apply this filter to the noisy curvature values assigned to each scan point in the 3D domain *before* generating a final 2D curvature map.

In the 3D point space, we define *adjacency point neighborhood rings* and assign subscripts as shown in Figure 4 for the case of hexagonal adjacency. Note that only the inner two rings are shown, but that additional rings may be used. The first subscript identifies the ring and the second subscript identifies each point within a ring. The concept of neighborhood rings applies similarly to orthogonal adjacency where, of course, the number of points in each respective ring will be greater.

We calculate the 2.5D Gaussian filtered mean curvature \hat{H} at each point P on the surface (indexed in turn as $P_{0,0}$) as follows:

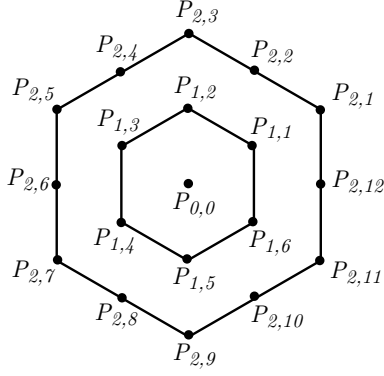


Figure 4: Adjacency point neighborhood rings.

$$\tilde{H}(P_{0,0}) = \frac{\sum_m \sum_n (w_{mn} H_{mn})}{\sum_m \sum_n w_{mn}}$$

$$\text{with } w_{mn} = e^{-(P_{mn} - P_{0,0})^2 / 2\sigma^2} \quad (3)$$

where the values of H_{mn} are the unfiltered mean curvatures. The summations are computed over m neighborhood rings with n points in each ring². Note that $\|P_{mn} - P_{0,0}\|$ is the physical (Euclidean) distance from the center point to a point in a neighborhood ring, and that the values of w can be thought of as representing filter weights. The standard deviation σ is smoothing factor that, in contrast with the 2D Gaussian filter, is now based on (estimated) surface area. The standard deviation retains its usual meaning in that we would expect, for example, the sum of the filter weights assigned to points within a two-sigma radius to be 95.5% of the total filter weight.

We refer to this as a 2.5D filter because it is computed on a 2D surface which is embedded in 3D space. Although there is some similarity to the standard 2D filter, note that, strictly speaking, this is not a convolution process and we have lost the computational efficiency of a fixed convolution kernel.

5 Experiments

We have performed experiments using scan data of the David statue from the Digital Michelangelo Project [1]. This data set was acquired with hexagonal point adjacency and thus neighborhood rings as illustrated in Section 4.2 were assigned for the computations. The data contains a moderate level of noise, with the equivalent Gaussian noise level standard deviation being approximately equal to the minimum scan point adjacency distance.

We concentrate on a curl of hair above David's right eye which only just visible at the top edge

²Including the center point $P_{0,0}$ as the single element in ring zero.



Figure 5: A photograph of Michelangelo's David.

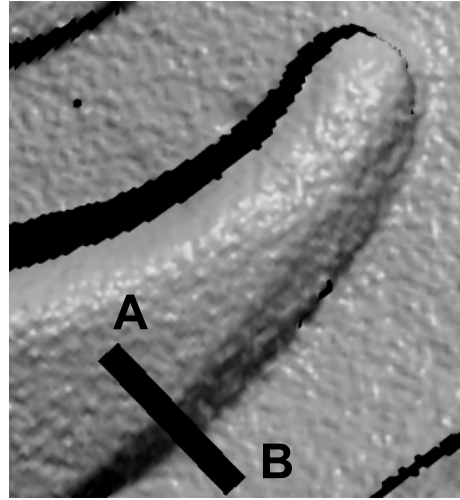


Figure 6: The curl: rendered mesh.

of the photograph shown in Figure 5. Figure 6 is a reference closeup of the curl rendered as a mesh surface [11] constructed from the raw scan data points. The lighting direction in this rendering has been chosen to highlight certain contours. The holes in the scan are due to occlusion. The black bar with ends marked A and B identifies a cut through a folded section of the surface.



Figure 7: Cross-section of curl scan points.

Figure 7 shows a cross-section slice of the scan points which indicates a folding edge between locations A and B. Traversing from A to B, the curvature starts as slightly positive, is distinctly positive

at the first bend, is then zero, is distinctly negative at the second bend, and then slightly negative.

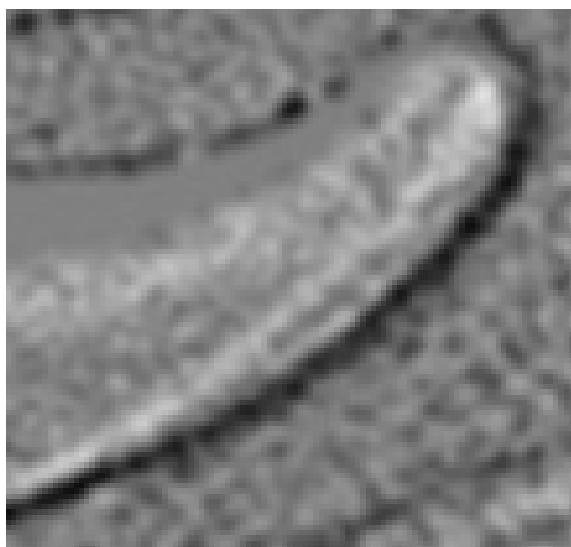


Figure 8: Curvature map: 2D filtering.

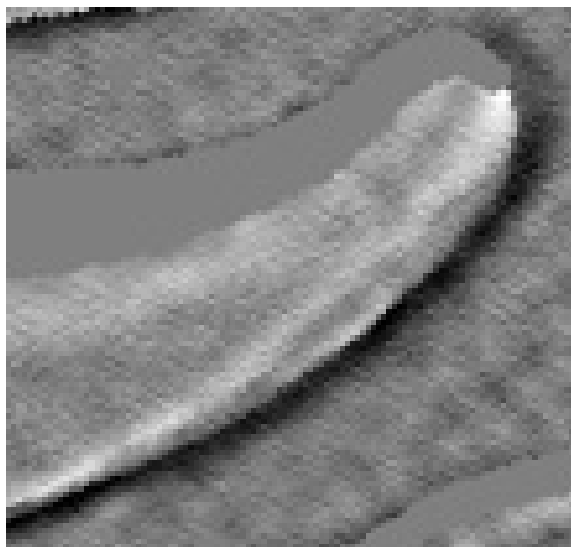


Figure 9: Curvature map: 2.5D filtering.

Figure 8 shows a shading encoded mean curvature map of the curl with 2D filtering. Maximum positive curvature is shading coded as white. Zero curvature is shading coded as medium grey and maximum negative curvature is encoded as black. Note that, because the filtering was done after the squashed dot mapping, there is some spread into regions where there is otherwise insufficient point data for curvature calculation. Figure 9 shows a shading encoded mean curvature map of the curl with 2.5D filtering. Equivalent values for the smoothing factor σ were used in both filters.

In Figure 9, it does appear that, as expected, there exists sharper transitions between black and white at the sharp fold edges. To confirm this, we extracted pixels associated with the previously illus-

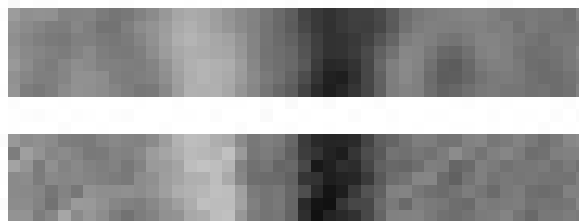


Figure 10: Extracted A-B cut pixels: 2D filter (top), 2.5D filter (bottom).

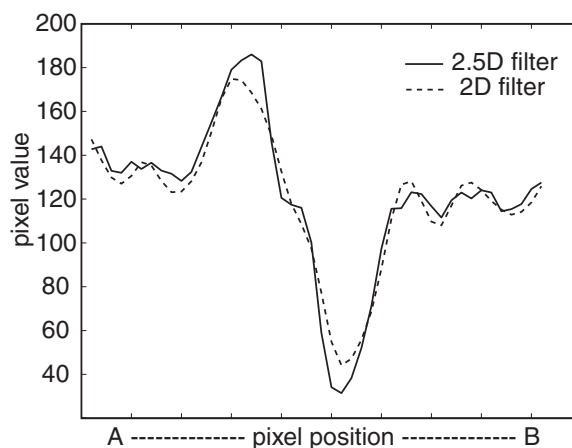


Figure 11: Pixel values along the A-B cut.

trated A-B cut from both the 2D and the 2.5D filtered curvature maps. The extracted pixels are shown in Figure 10.

We averaged the pixel values across the narrow cut direction and then plotted the resultant values as shown in Figure 11. In the center section of the plot, we see that, in the 2.5D trace, 1) the maximum slope is greater, 2) the magnitudes of both the positive and the negative peaks are greater, and 3) there is a distinct zero curvature shelf.

6 Conclusion and Further Work

In the experiment presented, 2.5D filtering results in more representative curvature at a fold edge than does 2D filtering. Further work is anticipated to include additional noise models (such as highly impulsive), additional filtering methods, and additional visualization techniques.

Finally, to illustrate the total size and scale of the scan, Figures 12 and 13 each show a curvature map of the entire scan, one with 2D filtering and the other with 2.5D filtering. Close examination reveals that other regions, such as the eyelid near the corner of the right eye, for example, appear to benefit from improved edge preservation.

7 Acknowledgements

The author would like to acknowledge financial support from The Department of Electrical and Computer Engineering at Manukau Institute of Technology. Access to the Digital Michelangelo Project data is courtesy of Stanford University.

References

- [1] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk, "The digital Michelangelo project: 3D scanning of large statues," in *Proc. SIGGRAPH*, pp. 131–144, 2000.
- [2] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *ICCV*, pp. 839–846, 1998.
- [3] M. Desbrun, M. Meyer, P. Schrder, and A. Barr, "Implicit fairing of irregular meshes using diffusion and curvature flow," in *Computer Graphics*, no. 33, pp. 317–324, SIGGRAPH 99 Proceedings, 1999.
- [4] S. Fleishman, I. Drori, and D. Cohen-Or, "Bilateral mesh denoising," in *ACM Transactions on Graphics 22*, no. 3, pp. 950–953, Proceedings of ACM SIGGRAPH, 2003.
- [5] A. Davies and P. Samuels, *An Introduction to Computational Geometry for Curves and Surfaces*. Oxford: Oxford University Press, 1996.
- [6] R. Klette, R. Kozera, L. Noakes, and J. Weickert, *Geometric Properties for Incomplete Data*. Dordrecht: Springer, 2006.
- [7] R. Klette and A. Rosenfeld, *Digital Geometry*. San Francisco: Morgan Kaufmann, 2004.
- [8] L. Alboul and R. van Damme, "Polyhedral metrics in surface reconstruction," in *The Mathematics of Surfaces VI* (G. Mullineux, ed.), (Oxford), pp. 171–200, Clarendon Press, 1996.
- [9] J. Rugis, "Surface curvature maps and Michelangelo's David," in *Image and Vision Computing New Zealand 2005* (B. McCane, ed.), pp. 218–222, 2005.
- [10] S. Hermann and R. Klette, "Multigrid analysis of curvature estimators," Tech. Rep. CITR-TR-129, Centre for Image Technology and Robotics, University of Auckland, 2003.
- [11] J. Foley, A. vanDam, S. Feiner, and J. Hughes, *Computer Graphics: Principles and Practice*. Boston: Addison-Wesley, 1996.

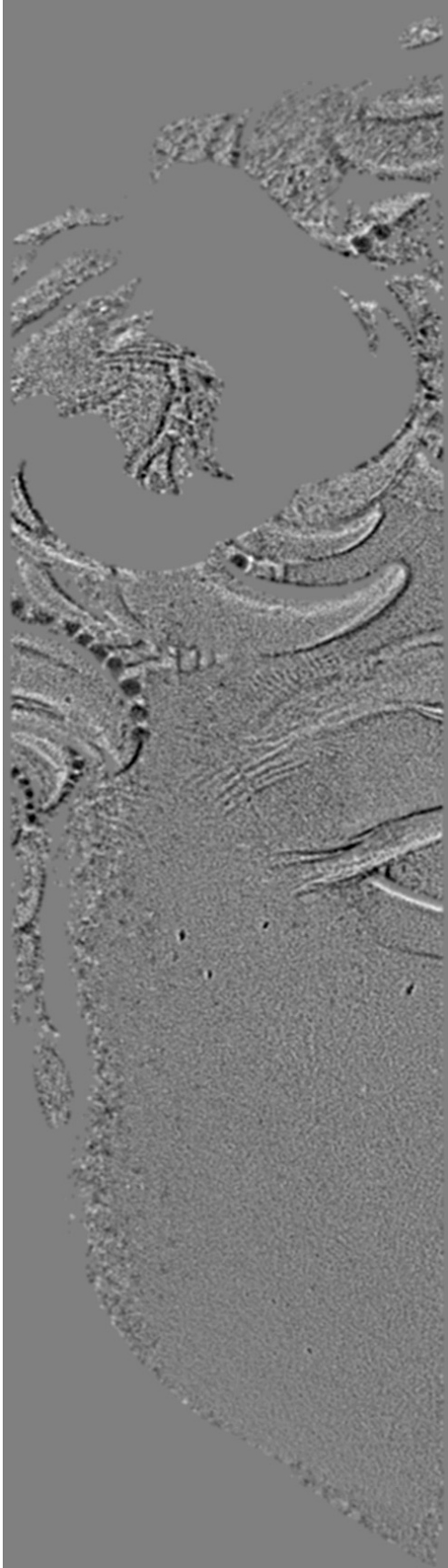


Figure 12: David's face: 2D filtering.



Figure 13: David's face: 2.5D filtering.

Occlusion Removal in Image for 3D Urban Modelling

E. H. Lim and D. Suter

Institute for Vision Systems Engineering,
Monash University, Victoria, Australia

Email: eehui.lim@eng.monash.edu.au

Abstract

In urban modelling, image occlusion can cause problems in the visualisation of the texture mapped 3D models, and in the analysis of point clouds. To resolve this issue, a popular strategy in image occlusion removal [1], given two or more input occluded images with same viewpoint, was studied and improved. Our revised technique is capable of removing regions in overlapped occlusions that have only been seen once. The algorithm can detect occlusion that is not possible to be resolved and reporting to the operator that manual intervention may be required by analysing the pixel values of the images at the internal and external boundaries of the grouped occluded regions. The performance of the algorithm was validated using the collected images from our calibrated camera on a Riegl laser scanner.

Keywords: Occlusion removal, terrestrial image, urban modelling, consensus image, boundary difference

1 Introduction

Accurate 3D surface modelling in urban areas is essential for a growing number of applications such as disaster management and environmental simulations. Other applications include regional planning, virtual reality and simulation of the propagation of radio waves for the cell phone industry.

Traditionally, urban simulation models can be obtained by processing data from photogrammetry. LIDAR (Light Detecting and Ranging) data is a relatively new method to obtain data for urban models. The use of a “Multisensor” – laser scanner and camera, permits a much faster, more complete and more efficient data acquisition. The laser scanner provides geometry data whereas the image taken provides colour information for realistic texture mapping and is useful for further point cloud analysis.

One of the problems in the data acquisition is anomalous occlusion due to moving humans and other objects. There exist a distinct time difference between the data acquisition from the laser scanner and the camera. This causes anomalous occlusions to occur where moving object does not coincide at the same location on both the laser scan and the image.

To illustrate such problem, refer to the following scenario: as shown in figure 1, a person moved from A, when the laser scan was taken on the grass plane, to B, when the image scan was taken that may occur at a time before or after the laser scanning took place. This caused anomalous occlusions to occur, as shown in figure 2 where the results for the image and laser

data were combined using the RiScan pro software (software from Riegl).

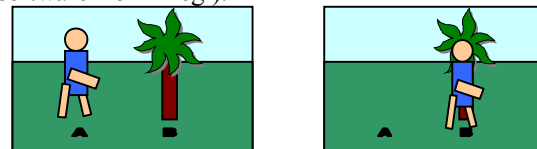


Figure 1 Example of where the scan artefacts occurs



Figure 2 Scanned results of occlusion occurs (a) Human object mapped with grass (b) Human image mapped onto tree trunk

The human object captured by the laser scanner in position A was colour mapped with grass, which is the image of position A after time α . On the other hand, the human image taken at time α was colour mapped onto the tree trunk and onto the ground on position B. Apart from being “unrealistic”, such false data collection can be a problem for further analysis. For example, if point cloud classification is based on the colour property, the green human object can be recognised as vegetation instead.

¹ α is the time taken from the laser scanning at position A until the laser scanning process completed plus the time taken for the camera image taking at position A, which is approximately the total laser scanning time

In general, there are two types of occlusions that have to be addressed: the occlusions in the LIDAR data and the occlusions in the image data. The occlusion in the LIDAR data can usually be solved by taking more than two scans in the same scan location and taking the greatest depth of each point, assuming the object moves fast enough with respect to the laser scan cycle.

To resolve the occlusions in the image data, the methods typically used are based on the idea that if two or more images contain views of the same scene at a different time, an unoccluded image can be formed with median filtering. Hence for each (i,j) location, the final image will be assigned the median of the RGB values (over that location in all images). However, this method performs poorly in very busy environments where the scenes are occluded more than 50% of the time. Moreover, it does not take any continuity properties into account. Therefore some detected occlusion might only be partially removed. A more sophisticated method for image occlusion is needed.

2 Previous Work

Ulm [2] removed the obstacles like cars or trees on terrestrial images by manual retouching of the artefacts or occlusions in a single image. However, this is very difficult and tedious.

Occlusions can also be eliminated via background modelling [3-5]. This is often used in visual tracking and surveillance system, where a long stream of video is taken from the same standpoint to initialise the background model with robust statistical methods such as the median. Wang [6] proposed a solution that locates all “stable subsequences” of pixel values in the video stream followed by choosing the most “reliable” subsequence with RANSAC. The initial background model then carries the mean value of the intensities over that subsequence.

However, we require a technique that does not need a large stream of images and is less computationally expensive. Our proposed algorithm is based on Herley's finding [1] that shows multiple images (>2 images) are not always necessary in solving image occlusion. As long as each location of the image is unoccluded at least once, it is possible to form an unoccluded image automatically. When the occlusion occurs there is generally a discontinuity around the boundary of the occlusion. The algorithm assumes that each connected set can be filled with data from a single connected set, and hence the problem is simplified to determining which image was the best. This works by comparing the similarity of the occlusions outer boundary in the consensus image

with the occlusions inner boundary in all the input images.

Herley's algorithm assumes that the occlusions are all independent objects; one occlusion boundary cannot consist of occlusions from different images. However, this assumption is often violated in our image acquisition in a busy environment. To remedy this, we improved the implementation in [1] to include the removal of occlusion when a single occlusion boundary requires information from more than single image.

In addition to that, we included the ability to detect unremoved occlusion. In the case where complete occlusion removal is not possible (which occurred in parts that are occluded in all input images), the algorithm is capable of detecting such case and perhaps shape retrieval or manual retouching can be done to recover the image. This is important as the number of the images in the acquired urban image database is large and it can be very time consuming to look through all processed images to select out images that need to be further processed.

3 Methodology

Let the images $I_0(i,j), I_1(i,j) \dots I_{N-1}(i,j)$ be the input images obtained from the calibrated camera taken at different time with same view point. Therefore $I_m(i,j) = I_n(i,j) \forall m,n$ unless either I_m or I_n is occluded at that location or affected by illumination changes.

The six steps of the algorithm are detailed below:

1. Construct consensus image

The consensus image U which carries visual similarity can be constructed from two or more images (acquiring pixel values from any two images that have difference less than threshold α) [1]. However, a simple way to remove occlusions for $N-1 > 2$ images would be working with two images (I_m and I_n) through the six steps and repeat with the resulting image (from I_m and I_n) and the third image up to the n^{th} image.

$$U(i,j) = \begin{cases} \frac{1}{2}(I_m(i,j) - I_n(i,j)) & \text{If } |I_m(i,j) - I_n(i,j)| < \alpha \\ 0 & \text{Otherwise} \end{cases} \quad (1)$$

Each pixel in the two images I_m and I_n is compared with a threshold α , where α is a small value to allow some matching error. We set α to 5 in our experiment. If the similarity is low, the consensus image U is assigned pixel value of zero. Otherwise, the consensus image carries the average of the pixel value in image I_m and I_n .

The visual similarity can be measured using many features such as intensity, colour, gradient, contour, texture, or spatial layout. A popular choice for similarity is colour due to its simplicity; and robustness against scaling, rotation, partial occlusion, and non-rigid deformation.

However, due to the fact that RGB colour space is sensitive to the change of illumination, and an outdoor environment can not be controlled; employing RGB colour space is less effective. Moreover, the images at the same position are taken with a time difference of at least one minute and illumination may change a lot. To curb this side-effect, similar to [7], we employed normalized RGB space r, g, b (where $r = R/(R+G+B)$ and $g = G/(R+G+B)$ and $b=B/(R+G+B)$). Figure 3b shows an example of the input images in the normalised RGB space and figure 3c shows the consensus image.

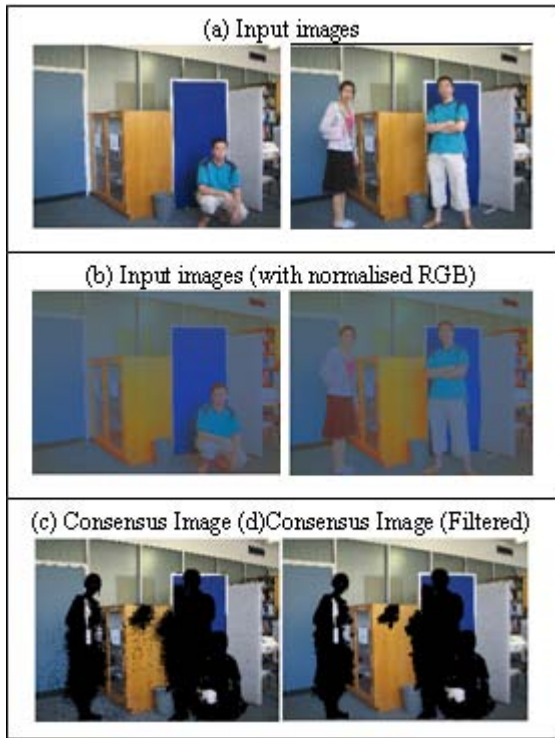


Figure 3 Construction of consensus image

2. Discard consensus image noise using low pass filtering

The consensus image at this stage may contain a large amount of occluded ‘holes’ (zero regions) due to true occlusions caused by moving objects and noise from illumination. In an outdoor environment, moving trees and bushes can generate a large number of small holes in the consensus image. Eliminating these relatively small occlusion ‘holes’ at this stage will reserve more computation time for the more complex processing part.

We employed a morphological filter which fills in the zero pixels that appear to be relatively small (less than 0.01% of the total pixels). It is important to select an image filter that does not change the position of the occlusion boundary (for instance, erode or dilate filter), as the accuracy of the true occlusion boundary has much effect on the occlusion removal. Figure 3c shows an example of the filtering result.

3. Form closed connected set in consensus image

Each occlusion which appears as a connected “zero pixel” region in the consensus image is grouped together as $S_p, p=1,2,\dots,P$, where P is the number of occlusion region in consensus image. For instance, in figure 3d, the consensus image has three ‘holes’ – $M=3$ and figure 4a shows an example of grouped connected zero pixels.

Similar to [1], for each set of S_p , the internal boundary of each occlusion, $B_{mp}, m=1,2,\dots,M$, where M is the number of input images, is defined as the set of pixels in the zero-connected region that has at least one neighbouring non-zero pixel with the value from the input images. Therefore, for each set of occluded images, there will be $m \times p$ internal boundaries. The green outlines in figure 4b-g are the examples of the internal boundary.

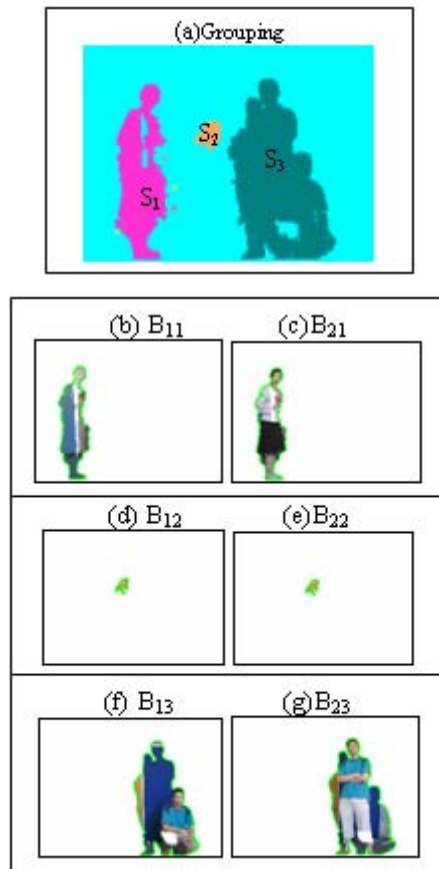


Figure 4 The occlusion boundaries

For each set of S_p , the external boundary E_p is defined as the set of pixels which is not in S_p and has at least one neighbour in S_p . The values of the external boundary are taken from the consensus image (or any input images, as the external boundaries are the pixels just outside the occluded region).

We need to provide a difference measure (step 4) for change across the boundaries. To do this, we calculate a single value for the external pixels of a single internal pixel. In this case we simply take the median of the external neighbours of an internal pixel.

For instance, in figure 5, the internal boundary = $\{a, b, c\}$ and the corresponding external boundary = $\{\text{median}(u,t,w), \text{median}(w,x,y,z), \text{median}(y,z)\}$.

		t	u
	x	w	a
	y	b	
	z	c	

Figure 5 Definition of external boundary

4. Test each set of S_p for occlusion overlap

A big difference in pixel value at the boundary is most likely to indicate an occlusion. Therefore, it is possible to recover an occlusion from pixels from a image provided that a small difference is constant throughout the boundary pair. Otherwise, it may indicate that it is insufficient for each S_p to only fill from a single image. In this case, S_p needs to be broken into n parts, where n is the number of actual occluding objects in one connected region. For example, in figure 4, S_1 can be filled from I_1 and S_2 can be filled either from I_1 or I_2 . On the other hand, S_3 has to be filled partly from I_1 and I_2 .

For each occlusion S_p , there is m external and internal boundary pairs. To determine where and whether the boundary needs to be divided, we analyse the trend of the difference of the boundary pairs. Let d_{mp} be the difference of internal and external:

$$d_{mp} = |B_{mp} - E_p| \quad (2)$$

In the case where S_p can be filled from single image, d_{mp} will constantly be relatively small for at least one m (image). For example, in figure 4, d_{11} will be constantly relatively small and d_{21} will be constantly relatively large; on the other hand d_{13} will be partially relatively small (where d_{23} is relatively large) and partially relatively large (where d_{23} is relatively small), which indicates filling from single image is insufficient. Therefore this implies that if there is no zero-crossing in D_p , where $D_p = d_{mp} - d_{mp}$, the set of S_p can be filled from single image.

If the number of zero-crossing in D_p is non-zero, we need to discover the dividing location in the occlusion

boundary, which can be obtained from the location of the zero-crossing in D_p . For instance, in figure 6a, $D_3 = d_{23} - d_{13}$ is plotted (smoothed with a moving median filter). The negative regions indicate filling from I_1 and positive region from I_2 . The location where the boundary needs to be broken apart (shown in figure 6b) can be discovered from the location of zero crossing (labelled as A and B).

5. Fill up consensus image zero regions

Each occlusion hole S_p which now only requires information from single image is filled with pixel values from the most suitable image.

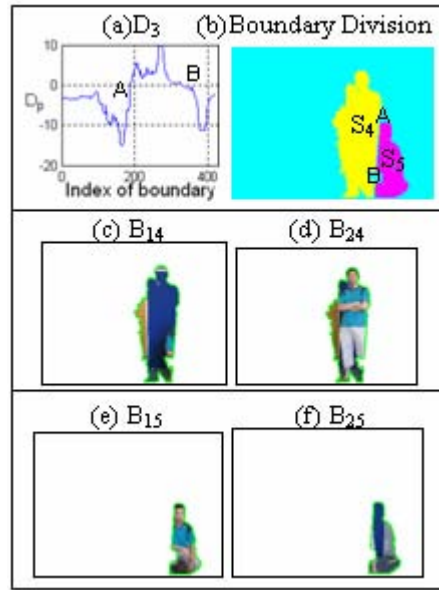


Figure 6 Boundary separation

The selection criterion is decided based on the degree of discontinuity, L , for each S_p in each image. When an occlusion occurs, the variation of the pixel values at the boundary of occlusion will be relatively large. Therefore S_p will select the image with lowest L to fill in the occlusion region.

$$L_{mp} = \sum_{i=1}^k d_{mp}(i) \quad (3)$$

where k is the number of pixels in the boundary.

Consider the example in figure 4b,c: $L_{11} = 16$ and $L_{21} = 49$. Therefore, S_1 will be filled from I_1 .

6. Blend and retrieve realistic shape

Blending [8, 9], which is often employed after image stitching, is required to provide a realistic final output in order to overcome the illumination and colour difference of the filled occluded region that can come from different images.

In the case where the overlapped region is too big to be entirely recovered (which is indicated by the

distance between the locations of boundary division) the algorithm is able to identify this situation and employs manual retouching to recover the final output. For instance, in figure 6b, the distance between A and B is relatively large and this confirms that the occlusion cannot be entirely removed. For full automation, a shape retrieval algorithm [10] can be implemented.

4 Results

A selection of our results is shown in figure 7, 8 and 9:

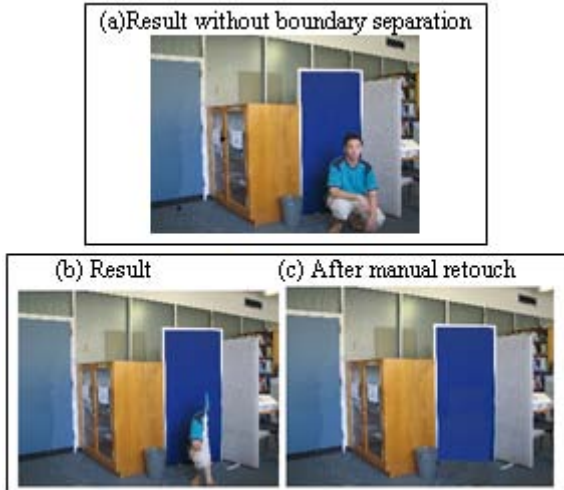


Figure 7 (a) Without the boundary separation, S_3 shown in figure 4a have to be filled from one image with the lowest b_p which is b_{13} . (b) With boundary separation, the best recovered consensus image still contains the occlusion at the unobservable background region. (c) The algorithm can detect the insufficient of background information and prompt for manual retouch or shape retrieval.

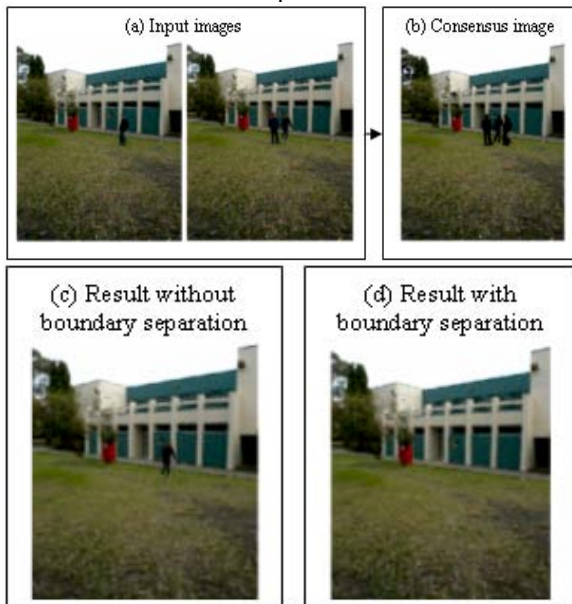


Figure 8: Input images with occlusions and result of the implementation (a) Input image sequence; (b) Consensus Image with occlusion “holes”; (c) Without boundary separation, the occlusions that overlapped is only filled from second image; (d) With boundary separation, all occlusions are removed automatically as occlusion overlapped at relatively small region.

Due to the reason that only the immediate pixel inside and outside the occlusion boundary is considered, the construction of a consensus image is important. We observed that shadows (which are not entirely removed in the normalised RGB space) are very

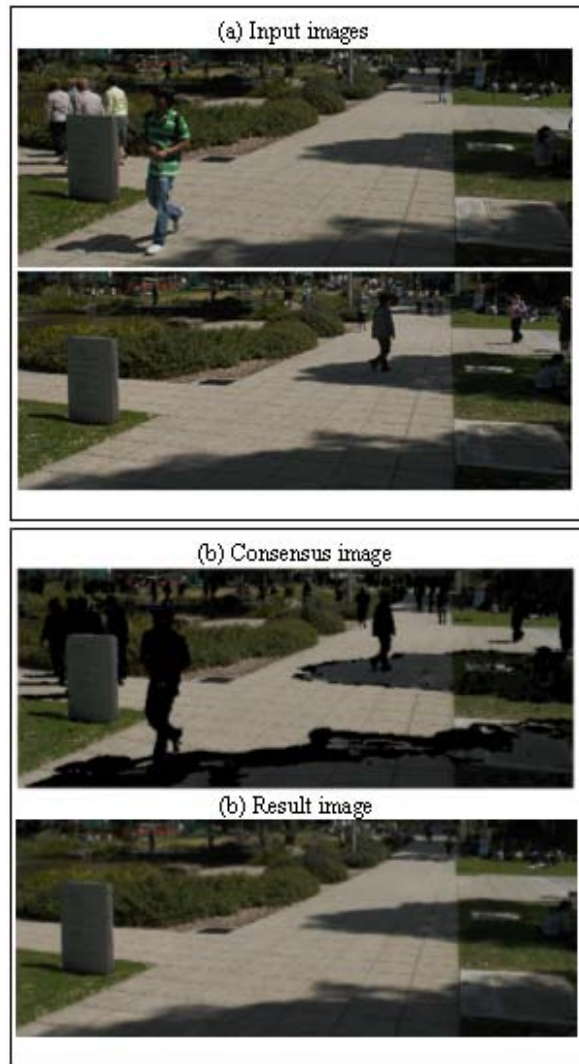


Figure 9: Input images with occlusions and result of the implementation (a) Input image sequence; (b) Consensus Image with occlusion “holes”; (c) Most occlusions are removed

likely to cause the boundary of the real occlusion to be detected outside the real occlusion location. This will affect the level of discontinuity when determining which input image has the “best” region to fill from.

5 Conclusion

We had improved the algorithm shown in [1] to include the capability of removing overlapped occlusions in input images and the ability to alarm unresolved occlusions. We tested the algorithm on the images collected from the calibrated digital camera of our laser scanner system. Most images were occluded with human or other objects such as cars and trolleys. The algorithm is capable of removing most of the occlusions unless the background cannot be observed in at least one of the images, for example when a car is parked throughout the entire data acquisition session. This is not possible to solve without additional background information. Relatively small occlusions (far from camera) with shadows are also more likely to be unremoved due to the inaccuracy of detected boundaries. Further work includes using the information of the pixels nearby the boundary, other than the immediate pixel adjacent to the boundary, to lower the effect of inaccurate occlusion boundary.

6 Reference

- [1] C. Herley, "Automatic occlusion removal from minimum number of images," *2005 International Conference on Image Processing*, vol. 2, pp. 1046-9, 2006.
- [2] K. Ulm, "City models from aerial imagery – Integrating images and the landscape," *GEoinformatics, January/February*, vol. 8, pp. 18-21, 2005.
- [3] B. Hongqiang and Z. Zhaoyang, "Identification of occlusion regions based on background rebuilding for automatic video object segmentation," *Proc. SPIE - Int. Soc. Opt. Eng. (USA)*, vol. 5286, pp. 883-6, 2003.
- [4] T. Li, W. Chengke, L. Shigang, and Y. Yaoping, "Complete structure recovery from long image sequence with occlusions," *Proc. SPIE - Int. Soc. Opt. Eng. (USA)*, vol. 5286, pp. 529-34, 2003.
- [5] B. B. Madhavan, C. Wang, H. Tanahashi, H. Hirayu, Y. Niwa, K. Yamamoto, K. Tachibana, and T. Sasagawa, "A computer vision based approach for 3D building modelling of airborne laser scanner DSM data," *Computers, Environment and Urban Systems*, vol. 30, pp. 54-77, 2006.
- [6] H. Wang and D. Suter, "A novel robust statistical method for background initialization and visual surveillance," *Lecture Notes in Computer Science, Hyderabad, India, Asian Conference on Computer Vision (ACCV)*, vol. 3851, pp. 328-337, 2006.
- [7] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proceedings of the IEEE*, vol. 90, pp. 1151-63, 2002.
- [8] H. P. A. Lensch, W. Heidrich, and H. P. Seidel, "Automated texture registration and stitching for real world models," *Proceedings the Eighth Pacific Conference on Computer Graphics and Applications*, vol. 1, pp. 317-452, 2000.
- [9] S. T. Y. Suen, E. Y. Lam, and K. K. Y. Wong, "Digital photograph stitching with optimized matching of gradient and curvature," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 6069, pp. 148-159, 2006.
- [10] M. Hoyneck and J. R. Ohm, "Shape retrieval with robustness against partial occlusion," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 593-6, 2003.

Modelling Interactions with a Computer Representation of the Upper Gastrointestinal System

Gastélum Alfonso¹, and Márquez Jorge¹

¹ National Autonomous University of Mexico, CCADET, laboratory of analysis of images and visualization, México

Email: valdenar@gmail.com

Abstract

Our goal is to simulate interactions with a computer model of the esophagus and the stomach, built from the *Visible Human* database, and reported in previous works. The optical distortion of the endoscope was simulated during navigation allowing the user a quantitative assessment of distortion when he measures an injury. Another improvement to our model is the inclusion of abnormal anatomy, from two types, the first one is related to the color features of the disease and the second is a direct modification in the triangle structure of the mesh, with the goal to simulate blisters and injuries in the model. The esophagus in its natural state presents radial collapsing, which was simulated using finite element methods. To this collapsed state we superposed the interaction of the triangle mesh with a model of the air pressure against the walls. The collapsed state allows to train the user in the insertion of the endoscope and to assess the effects of friction between the endoscope and the walls of the upper gastrointestinal system.

Keywords: SPH, optical distortion, mesh color diseases representation.

1 Introduction

Computer training in endoscopic procedures allows the specialist to interact with a virtual model and provides different points of view of the anatomical area of interest. Such enriched navigation permits the specialist to have a better understanding of the whole volume. To complete a computational training system from our upper gastrointestinal model, reported in [1], we developed a navigation environment that permits a user to explore the model and to be trained on anomaly detection.

Computer models for endoscopy training have already been reported [2]; our approach is to give the user the possibility of train in two areas that these models do not provide: the optical distortion and the insufflation process in the stomach.

The goal of the present work is to simulate interactions with a computer model of the esophagus and the stomach, built from the color anatomical slices of the *Visible Human* database [3]. To simulate interactions with the model, the esophagus is collapsed to obtain a natural state. Navigation was also enhanced to allow the user to insufflate into the upper gastrointestinal track. These and other improvements provide the user with a closer experience to reality when he is training in the computational environment of our system.

2 Procedure

Since our model is built to serve as a training system, the navigation must provide the closest possible realistic behaviour, hence, we introduced an optical barrel distortion [4,5] that allows to experiment the point of view of a real endoscope. We also simulated abnormal anatomies and the deformation of the High GI system due to an insufflation process.

There are two main problems in this stage: the first is the computational burden of real-time deformation of the mesh, the second is related to the physical behavior of fluids and its representation in a computational environment. A problem arises when the process of air insufflation is introduced to our model: the increased time in rendering and a slow feedback interaction. To solve this we first divided the process of navigation and interaction into pre-calculated and real-time computed features, the first one are modifications to the color or the structure of the triangle mesh; these modifications are pre-calculated since their characteristics will remain the same during the time the user interacts with the model, the second one consists of mesh modifications during navigation, depending on user interaction, and cannot be pre-calculated. The real-time calculations use approximated numerical solutions, maintaining a fast interactive environment. The first step is to classify all mesh modifiers that do not need to be made in real time: these are color modifications and solid structure modifications.

2.1 Color Modification

There are two main changes made over the original model, the first one is the introduction of color obtained from video-endoscopies of healthy esophagus to replace the original color that represents our model, this is a necessary step in the improvement of the model because the original color was obtained from the VHP data base, where color is altered by the postmortem condition. In the Fig. 1 presented the result of the color change.

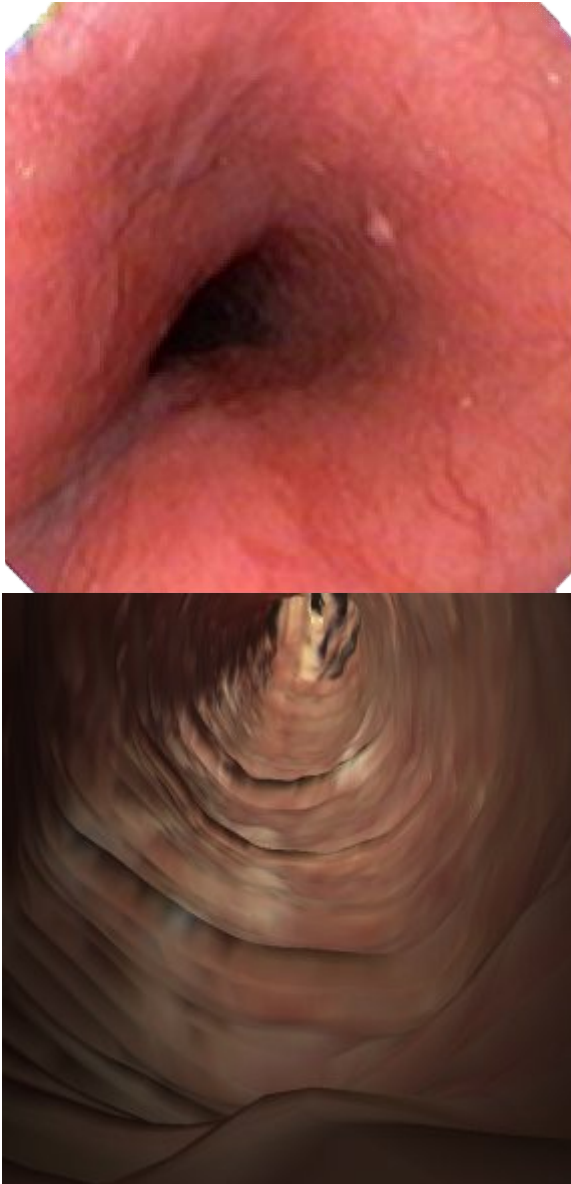


Figure 1: a) Real endoscopy, b) Model of the esophagus mapped with the color from the VHP database.

The next color modifications are introduced as a representation of diseases produced by the constant exposure of the esophagus to the peptic acid. This

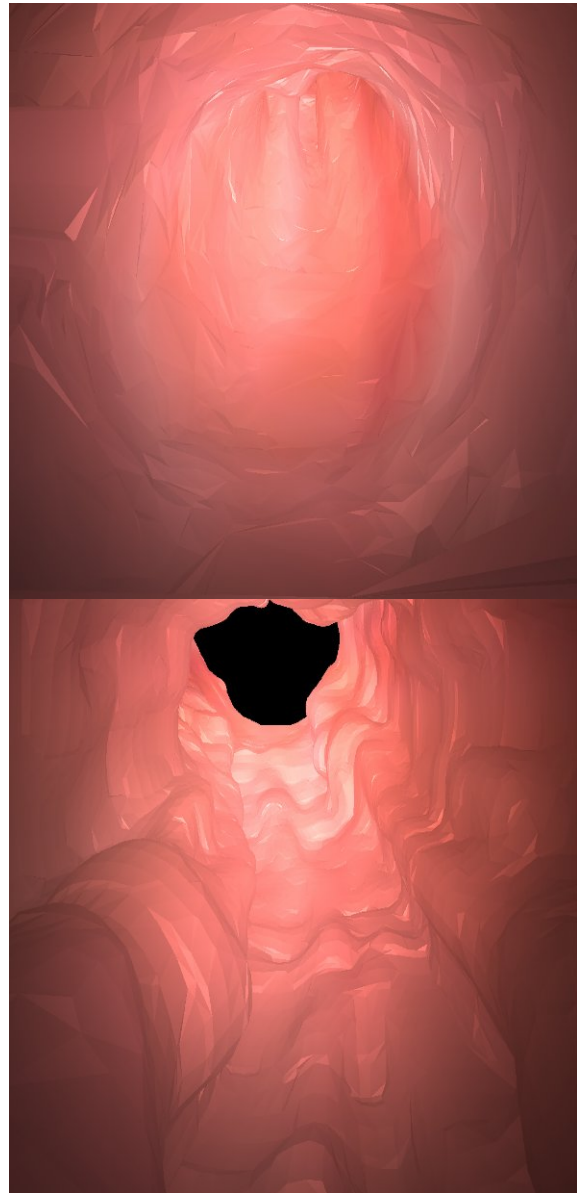


Figure 1 (continuation): c) Model with the color mapping from the real endoscopy, d) another view of the model

disease can be detected during the endoscope exploration; the physician must identify the color pattern of the affected zone and give an accurate measurement of the interest area

Such diseases can be considered to be in the initial stages before structure alteration, examples of such diseases are: Gastro-esophageal Reflux Disease (GERD), reflux esophagitis, Barret disease and some ulcers. [6]

In Fig. 2 is shown the effect of the peptic acid in the color of the model, depending on the type of disease and its severity. The intensity, hue and extension will change, whereas position and severity vary each time the model is run.

Colors for the model were obtained from video-

endoscopies from patients with a healthy condition or with some disease. At first, we considered the ideal case where the esophagus is represented by a cylinder and the center of this cylinder in XY plane gives us the position of the lens. Fig. 3 shows a visual representation of this cylinder.

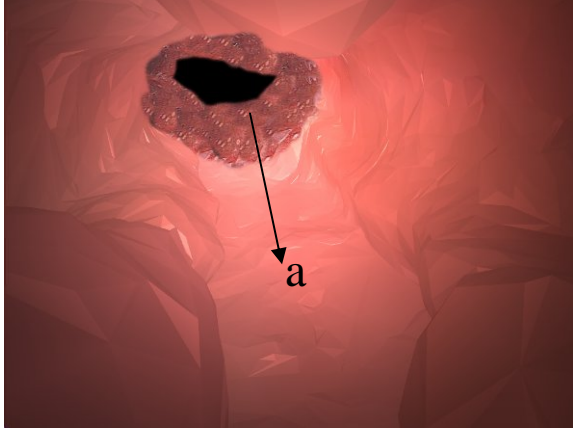


Figure 2: a) Barred disease present in the model of the esophagus.

This representation is useful for a first approach to color segmentation, the properties of the field of view of the wide angle lens are used to construct the mapping from the 2D images to 3D vertex. We consider that the greater the angle of view, the object gets closer to the lens in the Z axis (Fig. 4).

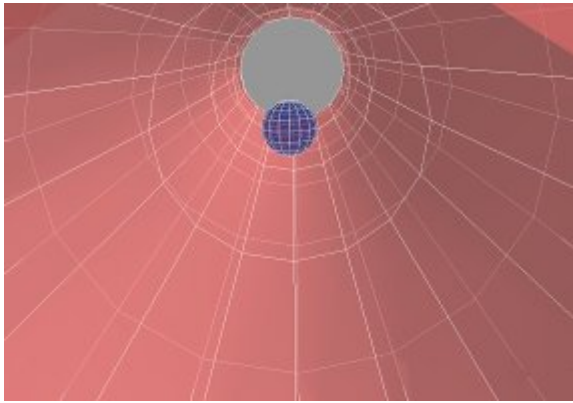


Figure 3: a) Ideal representation of the esophagus. The blue sphere represents the lens position.

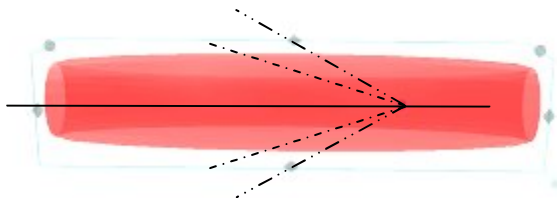


Figure 4: a) Segmentation of the color depending on the angle of view.

From the Fig. 4 we obtain that $\theta \approx Z$ depth. As the angle θ increases, the object gets away from the lens.

The figure 5 is a snapshot from a video-endoscopy.



Figure 5: Barrett disease, the image shows the *Esophageal-Stomach Junction*

The Fig. 5 shows a real endoscope image, where three problems are present:

- 1.- The center of the anatomical structure (esophagus stomach junction) and the center of the lens are not equal. To correct this, we introduce a difference Δx .
- 2.- To obtain a certain perception of depth, the esophagus-stomach junction is manually segmented and considered to be the farthest object in the image.
- 3.- The presence of folds is the most difficult problem related to segmentation, because we can not characterize 3D folds with only one image.

To map the color value from the images to the vertex table, the distance from the correct center to the interest pixel is calculated; all the pixels having a certain distance will belong to a contour with the same z value.

$$d = \sqrt{(x + \Delta x)^2 + (y + \Delta y)^2}$$

From each representative distance we obtain a list of pixels, whose number N would highly vary depending on the distance d to the center. The contour closest to the center would have less pixels, having this in mind, it is necessary to interpolate more color values because the N pixels are less than the number of vertex corresponding to the 3D model for that slice.

For each vertex the corresponding color value is obtained from the color value of the neighbors by bilinear interpolation.

The distances from the lens to the farthest object in the image in the z axis is obtained with the relation:

$$D_{\max} = \frac{r_{\max} - r_{\min}}{m}$$

Where m is a scale factor obtained from the size correction of the projection of a fix anatomical area, in the case of the GERD or the Barret diseases, the anatomical area is the “esophageal-gastric junction”:

$$m = \frac{\text{Real size}}{\text{Projection size}}$$

The number of discrete distances “ d ” necessary to map all the images to the model and the distance between them is obtained with the relation of D_{\max} and the scale of the z coordinate of our 3D model.

$$N_d = \frac{D_{\max}}{Z_{\text{scale}}}$$

$$\Delta d = \frac{D_{\max}}{N_d}$$

2.2 Solid Structures

The solid structure modifications are changes in the esophagus produced by extended damage. In our model we require to build new triangular meshes and a 3D modeling software (Truespace 6) is used to build the varises 3D meshes. Fig. 6 shows a rendering corresponding to a high degree of varises and the model of one.

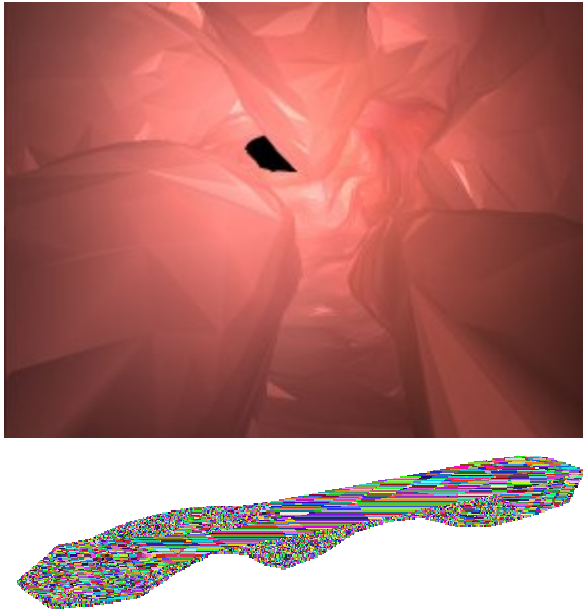


Figure 6: Superposing a mesh for varises over the original model.

The second kind of modifications are calculated in real time, depending on the user decisions and the position of the lens of the endoscope, so they required an algorithm preventing slow feedback; it also integrates the optical distortion and the simulation of the insufflation process during navigation.

2.3 Optical Distortion

A barrel geometrical distortion is present in common endoscopes. To obtain the distortion parameters in a real endoscopic lens we used a grid image viewed trough the lens. The two sets of points with and without lens are related by the following inverse transformation in implicit form [7]:

$$\begin{pmatrix} x_d \\ y_d \end{pmatrix}_{(lens)} = \begin{pmatrix} x_d \\ y_d \end{pmatrix}_{(obj)} - \left(\frac{\partial g((x_d, y_d), t)}{\partial ((x_d, y_d), t)} \right)^{-1} \left(g((x_d, y_d), t) - \begin{pmatrix} x_c \\ y_c \end{pmatrix} \right)$$

The result of the iterated process is shown in Fig. 3. The simulation of distortion is justified when a measurement is made. This modification is only applied during rendering where the original position of the vertex is not changed, and only calculated on visible features of the model.

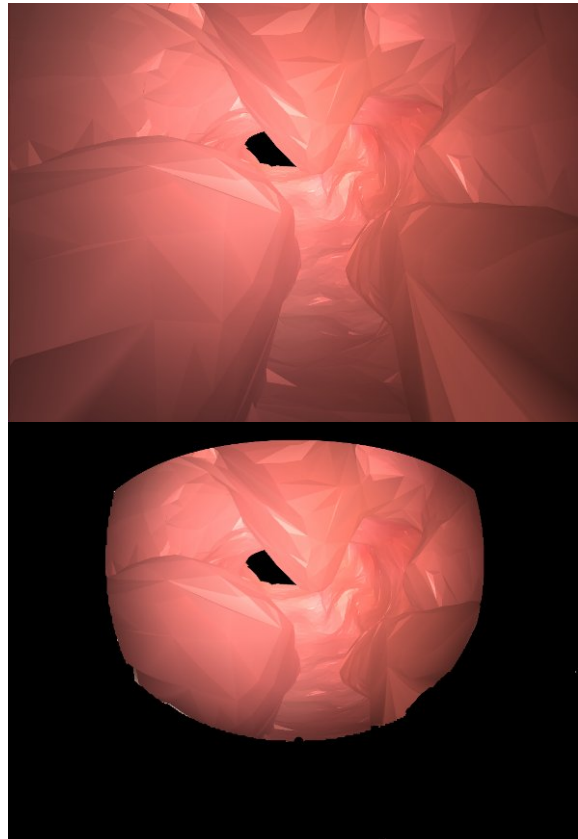


Figure 7: Change in the field of view due to the introduction of the lens distortion and the change of the angle of view to 100°

2.3 Fluid interaction

The high gastrointestinal system is collapsed at rest, hence, to improve the navigation and visibility, the endoscopist must introduce air in order to expand the esophagus and stomach. Therefore, we introduced the

simulation of such insufflation process.

To model this process we implemented the *Smoothed Particle Hydrodynamics* (SPH) fluid technique [8], using the properties of fluid dynamics to build an approximate behavior of the air pressure over soft tissue. Particle dynamics allows calculation of position and velocity of a set of particles and also to interpolate the properties of the air pressure in a given region, as well as the interaction with the boundaries.

Fluid dynamics is the study of fluid motion in response to forces such as gravity and pressure [9]. The equations of motion for a compressible fluid are:

$$\begin{aligned} \frac{d\rho}{dt} &= -\rho \nabla \cdot \mathbf{v} \\ \frac{d\mathbf{v}}{dt} &= -\frac{1}{\rho} \nabla P \\ \frac{du}{dt} &= -\left(\frac{P}{\rho}\right) \nabla \cdot \mathbf{v} \end{aligned}$$

Where ρ , \mathbf{v} , P and u represent density, velocity, pressure and energy, respectively.

The idea behind SPH is the determination of fluid characteristics by interpolating from a set of non-ordered points representing the particles. The fluid is partitioned into N regions with local densities defined by:

$$\rho_i = \sum_j m_j W_{ij}$$

Where m_j is the mass of the particle j , and the sum is over all particles. The interpolation is performed using a smoothing kernel W which is a weighted sum over particles within an area defined by a smoothing length h .

There are various forms of W , however the most advantageous (3) is:

$$W(r, h) = \frac{1}{\pi h^3} \begin{cases} 1 + \frac{3}{2}q^2 + \frac{3}{4}q^3 & \text{if } 0 \leq q \leq 1 \\ \frac{1}{4}(2-q)^3 & \text{if } 1 \leq q \leq 2 \\ 0 & \text{otherwise} \end{cases}$$

Where $q = r/h$ and r is the average distance between particles.

We can extend the idea of the kernel in SPH to the interaction with the boundary, in this case we consider a solid triangular mesh surface to be made of particles with an interaction kernel associated to them [10], so when the particles of the fluid enter their interaction

zone, defined by W , the pressure associated to the fluid particle is calculated on the vertex of the mesh. Fig. 8 shows the result of applying this equation.

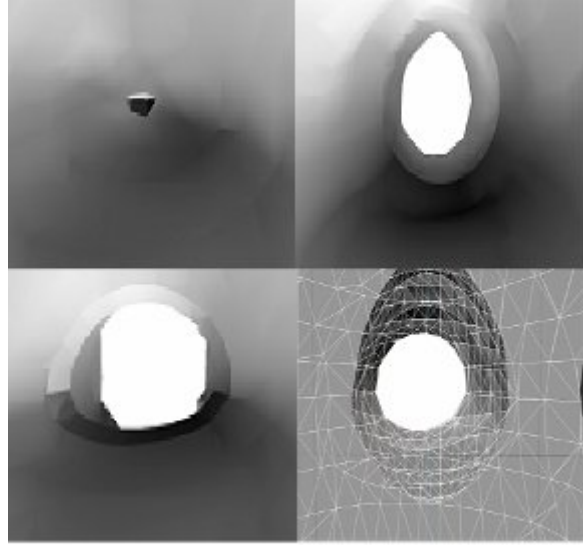


Figure 3: Result in the measure of the varices due to the optical distortion.

Figure 8: Insufflation process in a cylindrical model.

To simulate the wall-particle interaction, in the simplified cylindrical model, the propagation of the pressure force was calculated with the aid of finite-element and mass-spring methods.

3 Future Work

The next step in the construction of the training system is the integration of the FEM in the full High Gastrointestinal model (note that in the last paragraphs we described the application of FEM on a *cylindrical approximation*), in order to apply SPH to simulating the insufflation of the model in real time.

4 Conclusions

In order to obtain a computational training system for the gastrointestinal model that can be navigated in real time, we required an optimal combination of methods to simulate each of the behaviours and properties added so far. A first improvement was to pre-calculate everything that remains static during the navigation. The final model can be navigated with a high degree of interaction, since our rendering procedure employs a method that provided a discrete solution close to reality, and at the same time, fast enough to maintain a real-time feedback. In the case of optical distortion we modified the original position of the pixel on the screen into the corresponding distorted position; the distortion process is accelerated by pre-calculating, for each pixel, its final distorted position since the screen always maintains the same proportion. For the air dynamics we used *Smoothed*

Particle Hydrodynamics (SPH), achieving a realistic time response for user interactions with the model. Finally, when the process of insufflation is calculated, the walls of the esophagus expand accordingly.

5 Acknowledgements

We acknowledge the support of MS. Miguel Angel Padilla in the development of the finite-element and mass-and-spring methods for the mesh of our model. We also gratefully acknowledge Medical Doctor José Luis Mosso, for providing video-endoscopical material, advising and visual evaluation of model renderings.

6 References

- [1] Gastélum Alfonso and Márquez Jorge: "Construction of a model of the upper gastrointestinal system for the simulation of Gastroesophagoendoscopic procedures". VIII Mexican symposium on medical physics, American Institute of Physics 724, México, 2003, pp. 196-198.
- [2] Morten Bo-Nelsen, Joseph L. Tasto, Richard Cunningham, Gregory L. Merrill, "Preop™ endoscopic simulator: A pc-based immersive training System for bronchoscopy", HT Medical Systems inc, 6001 montrose road, suite 902, rockville, USA.
- [3] V. Spitzer, M. J. Ackerman, A. L. Scherzinger, and D. Whitlock, 1996, The Visible Human Male: A Technical Report, J. of the Am. Medical Informatics Assoc., 3(2) 118-130.
- [4] Tsai RY, "A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses". IEEE J Robotics vol. 3. p. 323-344, 1987.
- [5] F. Devernay and Faugeras, "Straight Lines Have to Be Straight: Automatic Calibration and Removal of Distortion from Scenes of Structured Environments", Machine Vision and Applications 1, p. 14-24, 2001.
- [6] ASGE Publication No. 1006, "THE ROLE OF ENDOSCOPY IN THE MANAGEMENT OF GERD" American Society for Gastrointestinal Endoscopy, Manchester, MA, March, 1986
- [7] Gergely Vass and Tamás PerlakI: "Applying and removing lens distortion in post production", Second Hungarian Conference on Computer Graphics and geometry 2003, Budapest, speech. www.vassg.hu/pdf/vass_gg_2003_lo.pdf
- [8] J.J. Monaghan, "Smoothed particle hydrodynamics", Ann. Rev. Astron. Astrophys. 30 (1992) 543-574.
- [9] J. J. Monaghan and J. C. Lattanzio, "A refined particle method for astrophysical problems". A&A, 149, 135-143. (1985).
- [10] Matthias Muller, "Interaction of Fluids with Deformable Solids", ETH Z'urich, Switzerland

Acquiring Visual Hulls by Voxels

Yu-xuan HONG and Richard Green

³Department of Computer Science, University of Canterbury

Email: yxh10@student.canterbury.ac.nz

Abstract

This paper describes a method for acquiring the visual hull of an object with known background. The first step of this method is acquiring sets of silhouettes from multiple photos of the object. The second step is using these silhouette images to construct the 3D model of the object (visual hull of the object). In the later part of the paper, the limitations of the proposed method are explored, and future possible improvements are also presented.

Keywords: Visual hull, 3D model, silhouette, voxel, computer vision.

1. Introduction

More and more 3D computer graphics are used in modern electronic games, movies and real world simulations. Better realism, visual effect and simulation effect allow 3D computer graphics to take more places over traditional 2D computer graphics. To be able to construct 3D models of real world objects effectively has become an important topic.

Currently, there are laser-scanning systems for 3D shape acquisition. However, most of these systems are expensive. Also when they are used in acquiring 3D human body shape, much useful information is lost. Similarly, commercial marker-based motion capture systems are invasive and difficult to use. In applications such as security/surveillance and human-computer interaction, these systems are not applicable because placing markers on the person is either impossible or undesirable. In contrast, there are many advantages of using a vision-based approach for 3D shape acquisition. For example, cameras are low-cost, easily reconfigurable and non-invasive. Moreover, camera images contain both shape and color (texture) information of the object. Therefore, one of apparent most promising solution is visual hull.

In this paper, two popular visual hull algorithms are briefly explained in the section 2 (“Background”). The “Constructing visual hulls by voxels” section presents the method used for visual hull construction in this paper. The “Experiment Setup” section describes the requirement and setup of the method. In the following section “Result” provides the result of the experiment in a scientific manner. In the “Conclusion and Future Work” section, the paper reveals the limitation of the method and giving directions for future improvement.

2. Background

To use silhouettes to acquire 3D shape of an object was first introduced by Baumgart in 1974. He used four silhouette images to estimate the 3D shape of a baby doll and a toy horse in his PhD thesis. [3] Since then, there are various different variation of this method have been proposed.

Twenty seven years later (1991), Laurentini [4] coined the term Visual Hull (VH). It has been used by researchers for over a decade to denote using silhouette to acquire 3D shape of an object.

There are currently many different visual hull acquisition algorithms. However, almost all of them are improvements of two approaches. One is two-dimensional surface based, and the other one is three-dimensional volume based.

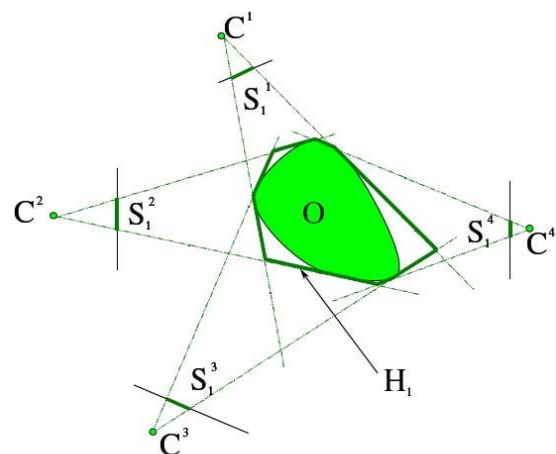


Figure 1: 2D surface based visual hull approach.[17]

A two-dimensional surface based method is normally performed in the following manner. From a view point p , there is a silhouette image S_p of the object. By allowing infinite rays emit from point p and passing through the interior points of S_p , we can obtain a cone-like volume C_p (visual cone). For every other view point p , there is a visual cone C . Intersecting all the visual cones, the visual hull of the object can be formed ($C_p = \cap S_p$). The visual hull is represented by the 2D surface patches which are obtained from intersecting the surfaces of the visual cones. This method is hard to implement and not computationally inefficient. When computing high resolution visual hulls, the method requires some form of CSG that increasing the complexity of the method more. Therefore, this method is rarely used in modern visual hull researches.

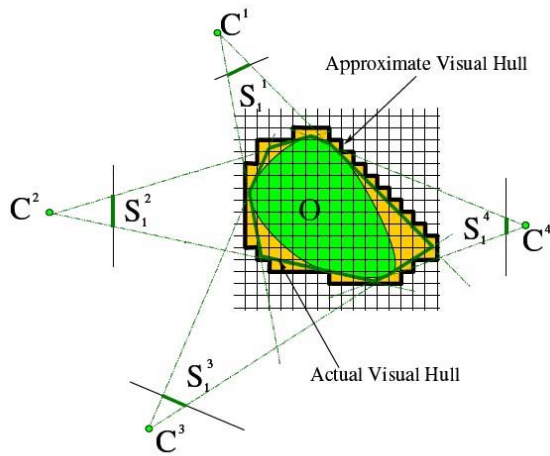


Figure 2: 3D volume based visual hull approach.[17]

Three-dimensional volume based method is developed after two-dimensional surface method. The method creates a cubic volume which is formed by a number of small cubes. Each cube is projected to the silhouette images of the object. If the projection of the cube lies outside of the silhouette images, the cube will be eliminated from the volume. After much iteration, the volume will form the visual hull of the object. This approach is relatively more efficient and accurate than the two-dimensional surface based approach. Most of recent visual hull researches are base on this approach, such as [5], [6], [7], [8], [9], [10].

An important fact is the silhouette based visual hulls can never be as accurate as the original object. This is due to the inability of identifying concave surface by silhouette.

3. Constructing Visual Hull by Voxel

3.1 Silhouette Extraction

Silhouette extraction is the fundamental step of the visual hull construction process. Its accuracy directly

affects the outcome of the visual hull. Therefore, using a precise silhouette extraction algorithm is vital.

This paper employs a robust algorithm using histogram information in the HSI color space to extract the silhouette of an object. The following diagram illustrates the process of the algorithm.

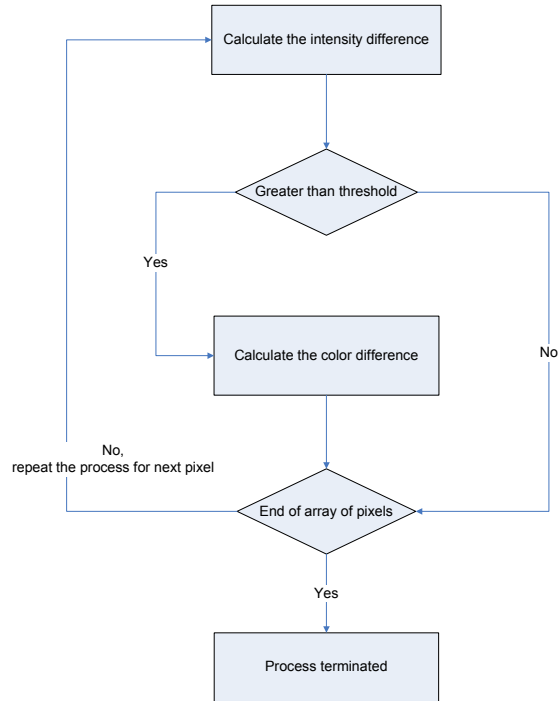


Figure 3: The silhouette extraction process

As the diagram shown, the algorithm precedes two major actions on each pixel of the picture. The first action is finding out the intensity difference of a temporary pixel in the current image and the background image. If the difference of the intensity is greater than a predefined threshold, the pixel will be set as a candidate of silhouette pixel. If the difference of the intensity is less than the predefined threshold, the algorithm will check whether there are more pixels need to be tested in the pixel array. So far, the process is still under first action. The goal of this action is extracting the difference between the known background image and current temporary image. However, this difference includes the object's shadow.

The second action of the algorithm calculates the color difference of previously obtained candidate pixel between the current image and the background image. If the color difference is less than a predefined threshold, the candidate pixel will be set as a real silhouette pixel, else it will be excluded. This action is excluding the shadow pixels which are set as part of silhouette in the first action. The calculation details in the algorithm are explained in the following two paragraphs.

As mentioned above, there are two major actions in this silhouette extraction algorithm. The first action calculates the intensity difference of a pixel by the following formula.

$$I_{DIFF} = \|I_{CT} - I_{BG}\|$$

I_{DIFF} indicates the difference of intensity, and I_{CT} and I_{BG} are the intensity of the current image and the intensity of the background image respectively. Every “ I ” is measured as a vector. The three components of the vector are the values of the pixel in RGB color domain. Sign “ $\| \cdot \|$ ” represents the norm of its content.

The second action which calculates the color difference is performed by following formula.

$$\theta = \cos^{-1} \left[\frac{I_{CT} \cdot I_{BG}}{\|I_{CT}\| \|I_{BG}\|} \right]$$

“ θ ” is the angle between the vector “ I_{CT} ” and “ I_{BG} ” in the RGB color domain. It is a measure of the color difference of a pixel between the current image and background image. As mentioned above, I_{CT} is the intensity of the current image, and I_{BG} is the intensity of the background image.

This silhouette extraction algorithm is relatively robust and reliable. Related experiment result is demonstrated in the “Result” section.

3.2 Visual Hull Construction

The visual hull construction method used in this paper is a three-dimensional volume based method. In the “Background” section, the three-dimensional volume based method has already been briefly explained. In order to understand this method in detail, the concept of voxel will be introduced here. A voxel is a “volume element” as that a pixel is a “picture element”. It is the smallest element in a three-dimensional volume. The algorithm presented in this paper eliminates the voxel which should not be included in the visual hull by processing through all the known silhouette images. The following pseudo demonstrate the process in detail:

1. Divide the interested space into $N \times N \times N$ discrete voxels $V_n, n = 1, \dots, N^3$.
2. Initialize all the N^3 voxels as visual hull voxels.
3. For $n = 1$ to N^3 {
 For $j = 1$ to J {

- (a) Project V_n into the k^{th} image plane by the projection function $\pi_j^k(V_n)$;
- (b) Eliminate the pixel V_n if the projected area $\pi_j^k(V_n)$ lies completely outside S_J^K .

}
 4. The uneliminated voxel forms the visual hull of the object.

In this algorithm, each voxel is projected from its real world position to the image position. The projection function $\pi_j^k(V_n)$ can be decomposed to following two equations.

$$x' = f' \frac{x}{z} \times M_{orientation}$$

$$y' = f' \frac{y}{z} \times M_{orientation}$$

“ x ” and “ y ” represent the coordinate of a voxel in a 2D silhouette image. “ f ” is the focal length of the camera which is used to take the silhouette images. “ x ”, “ y ” and “ z ” are the coordinate parameter of the voxel in the real world. The term “ $M_{orientation}$ ” is the current orientation of the voxel respect to the camera.

The focal length (f) of the camera and orientation of the object all can be acquired through a camera calibration process. Since camera calibration is not part of a visual hull acquisition algorithm, it is not discussed here. For further information, please refer to [18]. The values of “ x ”, “ y ” and “ z ” can be assumed, due to the exact size of a 3D object is not the main purpose of the visual hull, but shape.

By knowing above parameters, after applying the algorithm described in the pseudo code, the visual hull of an object can be formed.

4. Experiment setup

This section describes the required setup for the visual hull method proposed in this paper.

4.1 Camera calibration

The last section mentioned focal length and orientation need to be known to calculate the projected coordinate of a voxel on the silhouette image. In the experiment, those parameters are obtained by a camera calibration program developed by Danail Stoyanov [19]. Through the program, multiple photos of a chessboard are taken. These photos are inputs for acquiring the camera intrinsic parameters (focal length

and distortion coefficient of the camera). The focal length is used for voxel based volume reconstruction step later, and the distortion coefficient is used for obtaining correct silhouette image.

4.2 Silhouette acquisition



Figure 4: Setup of silhouette acquisition

Like the above figure shown, an object is placed on a home made turntable. (The turntable is underneath of the basketball.) There are two direct light source to ensure the result of silhouette acquisition. Multiple photos of the object are taken from different angles by rotating the turntable. Those photos are used as inputs for generating silhouette images by a program developed for this project.

4.3 Voxel reconstruction

A program developed for this project uses the silhouette images as input to execute the algorithm described in the “Visual Hull Construction” section. The output of the program is printed into a “wrl” file. The file is consisted of coordinates of the visual hull in 3D virtual world. It can be displayed by a VRML (Virtual Reality Modelling Language) browser.

5. Result



Figure 5: Silhouette which includes the shadow of the basketball.



Figure 5: Silhouette which excludes the shadow of the basketball,

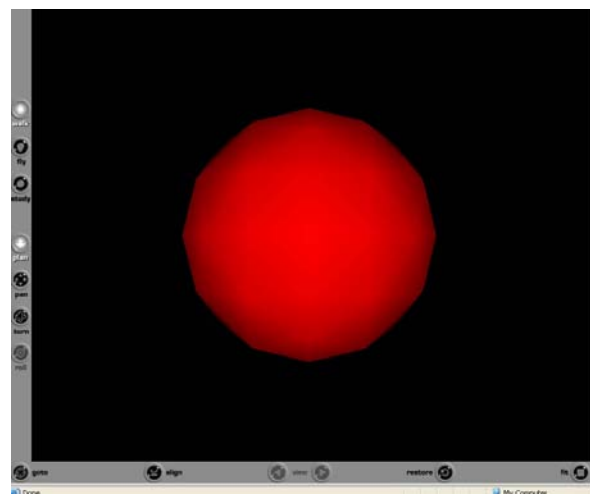


Figure 6: Visual hull of a basketball.

Table 1: The accuracy of the silhouette algorithm presented in this paper.

	Intensity only	Both intensity and color
Accuracy (average)	43%	98%

This accuracy is obtained by comparing the result from the silhouette algorithm and manually acquired silhouette image.

6. Conclusions and Future Work

The method described in this paper is relatively easy to setup and computationally efficient. However, the usability of the method is also limited due to the necessity of method, using multiple photos from a single camera, which implies that this method is not able to acquire visual hull in real time. In future, with multiple video cameras, the method can be improved to be used for real time visual hull acquisition. The silhouette algorithm used in this project requires the system to have knowledge of the background. This limits the method to being only used for object visual hull acquisition with known backgrounds. Unknown or dynamically changing backgrounds render object visual hull acquisition as not achievable with current silhouette algorithms. In future, the method can adapt more advanced background/foreground silhouette algorithms as in [1] to enable more robust visual hull acquisition. At last, the visual hull construction method in this paper can also use an octree data structure to improve the overall performance.

7. Acknowledgements

8. References

- [1] Gang ZENG and Long QUAN, *Silhouette Extraction From Multiple Images of an Unknown Background*, 2004
- [2] Yasemin Kuzu and Volker Rodehorst *Volumetric Modelling Using Shape From Silhouette*, 2001
- [3] B.G. Baumgart. *Geometric Modeling for Computer Vision. PhD thesis*, Stanford University, 1974.
- [4] A. Laurentini. *The visual hull : A new tool for contour-based image understanding*. In *Proceedings of the Seventh Scandinavian Conference on Image Analysis*, pages 993-1002, 1991.
- [5] Chris Buehler, Wojciech Matusik, Leonard McMillan *Creating and Rendering Image-Based Visual Hulls* 1999.
- [6] Peter Eisert *Reconstruction of Volumetric 3D Models* 2005
- [7] Andrew Fitzgibbon and Andrew Zisserman *Automatic 3D Model Acquisition and Generation Of New Images From Video Sequences* 1998
- [8] Yasemin Kuzu and Volker Rodehorst *Volumetric Modeling Using Shape From Silhouette* 2001
- [9] W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan *Computer Graphics Proceedings, Annual Conference Series, Pages. 369-374* 2000.
- [10] Kristen Grauman, Gregory Shakhnarovich, Trevor Darrell *A Bayesian Approach to Image-Based Visual Hull Reconstruction* 2003
- [11] Wojciech Matusik, Chris Buehler, Leonard McMillan, Steven J. Gortler *An Efficient Visual Hull Computation Algorithm* 2002
- [12] Martin Kampel, Srđan Tosovic, and Robert Sablamnig. *Octree-based Fusion of Shape from Silhouette and Shape from Structured Light* 2002
- [13] Phillip Milne, Fred Nicolls, Gerhard de Jager. *Generation of Visual Hull Models of 3D Objects* 2003
- [14] Aldo Laurentini. *How Far 3D Shapes Can Be Understood from 2D Silhouettes*. 1994
- [15] Gregory G. Slabaugh, W. Bruce Culbertson, Thomas Malzbender, Mark R. Stevens, Ronald W. *Methods for Volumetric Reconstruction of Visual Scenes*, 2003
- [16] Yang Liu, George Chen, Nelson Max *Visual Hull Rendering with Multi-view Stereo Refinement*, 2004
- [17] Kong Man (German) Cheung *Visual Hull Construction, Alignment and Refinement for Human Kinematic Modeling, Motion Tracking and Rendering*, 2003
- [18] Jean-Yves Bouguet, *Camera Calibration Toolbox for Matlab* http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/parameters.html 2006
- [19] Danail Stoyanov, *Camera Calibration Toolbox* <http://ubimon.doc.ic.ac.uk/dvs/index.php?m=581> 2006

Simulation of Medical Imaging Modalities - A Tool for Numerical Evaluation of Image Processing Algorithms

F. Uhlemann

Department of Computer Science, University of Auckland, New Zealand.

Email: imaging@uhlemannweb.de

Abstract

Evaluation of medical image processing algorithms proves to be a vital but difficult task because ideal reference data is hardly available. By simulating the most significant aspects of the imaging process realistic data can be generated from an ideal segmentation. Employing numerical measures to compare the calculated and the underlying ideal solution an objective evaluation can consequently be performed. The described software phantom allows to simulate multiple medical modalities with various parameters of the imaging, distortion and deformation processes. Currently various magnetic resonance imaging (MRI) sequences, computed tomography (CT), positron emission tomography (PET), single photon emission tomography (SPECT) and ultrasound (US) imaging are incorporated. Using the described numerical measures segmentation and deformation results can be automatically evaluated and compared.

Keywords: simulation, evaluation, medical imaging, software phantom, segmentation, registration

1 Introduction

Processing of images from different medical modalities is a rapidly growing field. Especially automatic segmentation, i.e. identifying and separating different regions, and rigid or elastic registration, i.e. locating corresponding points of structures in different data sets, is a challenging task.

Due to the nature of medical imaging there is hardly any subject specific reference data available. With respect to resolution, signal to noise ratio, distortion and representation of certain structures some modalities serve as a *gold standard* to which all other imaging techniques are compared. Nevertheless the underlying true geometry of the structures remains usually unknown because the object under investigation is an alive patient. But without this ideal reference data it is difficult to objectively evaluate the quality of a manually and/or automatically determined object shape.

There are various ways to create reference data for evaluation – each associated with certain advantages and disadvantages. Some examples are the well known visible human [1], the BrainWeb project [2, 3] or physical phantoms [4].

In this article the focus will be on the method of simulating realistic images by means of a so called software phantom because this does not require any real physical objects like patients or physical phantoms to be used and allows to generate images of different modalities easily while having control over all modeled imaging parameters and artifacts.

Even though this general approach has been known and used for many years it should be noted that many of the widely used test data sets do not represent the artifacts and noise of real imagery sufficiently (e.g. the classical Shepp-Logan phantom [5]). Therefore they do not provide a suitable basis for quantitatively estimating the performance of image processing algorithms under "real world conditions" but rather serve as a valuable tool to illustrate the general behavior of the tested method. These limitations will be tackled with the proposed software phantom.

2 Simulation of the Imaging Process

To be able to simulate the real imaging process a preliminary study of the involved deterministic and stochastic physical (sensors, amplifiers...) and numerical (reconstruction algorithms...) processes was carried out. Additionally modality specific differences were taken into account. To make implementation and calculation time feasible some assumptions had to be made. These include:

- object geometry can be described by regions of different characteristic gray values/textures
- homogeneous noise processes in the whole object and image domain
- most significant imaging processes can be modeled by linear filtering and forward/backward transformations.

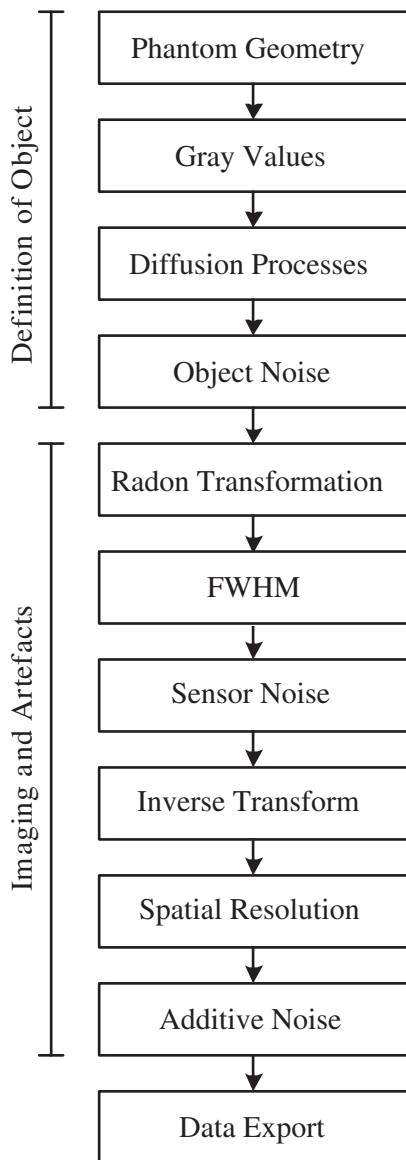


Figure 1: Processing steps of the software phantom's synthetic "imaging chain" without spatial transformations/distortions.

In figure 1 the identified and consequently modeled steps of the simulation process are shown. They will be explained in more detail in the following sections. See also figure 2 for a illustration of the simulation steps for a simple circular object.

2.1 Object Definition

The first step of the simulation is the definition of the object geometry by using a segmented image. There is no restriction on the possible shapes except that these should be sufficiently big relative to the image resolution to still be visible after being distorted by the consecutive imaging steps.

At this point it is possible to add a spatial transformation to the segmented image which allows to

create a series of deformed images for the evaluation of registration algorithms. Currently affine, sinusoidal and (smoothed) elastic deformations are available in the software phantom.

Following the assumption that objects should be differentiable and can be described by characteristic gray value or textures the next steps assigns these predefined gray value to the respective segment.

It should be noted that the observed gray values in medical images not only vary between subjects and body parts for the same class of tissue but that these usually also depend on the specific imaging device and the parameters used. For this work parameters from the BrainWeb project and other clinical data were used [4].

To be able to simulate the effects of infiltrative growth of tissues and pathologic structures (e.g. cancer) as well as the local perfusion of radioactive tracers (e.g. PET and SPECT) an averaging filter has been included to simulate these diffusion processes.

In the next step different noise distributions can be applied to the image gray values (Gaussian, Poisson, salt and pepper, speckle) to account for the stochastic nature of the image gray values.

2.2 Imaging and Artifacts

To create the projections of the object a Radon transformation is applied. In combination with a mean filter this models the limited resolution of the sensor array. The parameter to describe this effect is the "full width at half maximum" (FWHM).

Before the projections are reconstructed using an inverse Radon transformation noise can be added to simulate the effects of distorted sensor signals and/or quantum effects for nuclear imaging (PET and SPECT).

Finally the output resolution can be adjusted using interpolation and additive (unfiltered) noise can be added to the images.

2.3 Data Export

After the simulation has finished the created data along with the simulation parameters can be saved as a compressed MATLAB file or DICOM data. This allows to accurately document which data has been used for evaluation including the specific simulation parameters and to automatically annotate the final graphical presentation of the results.

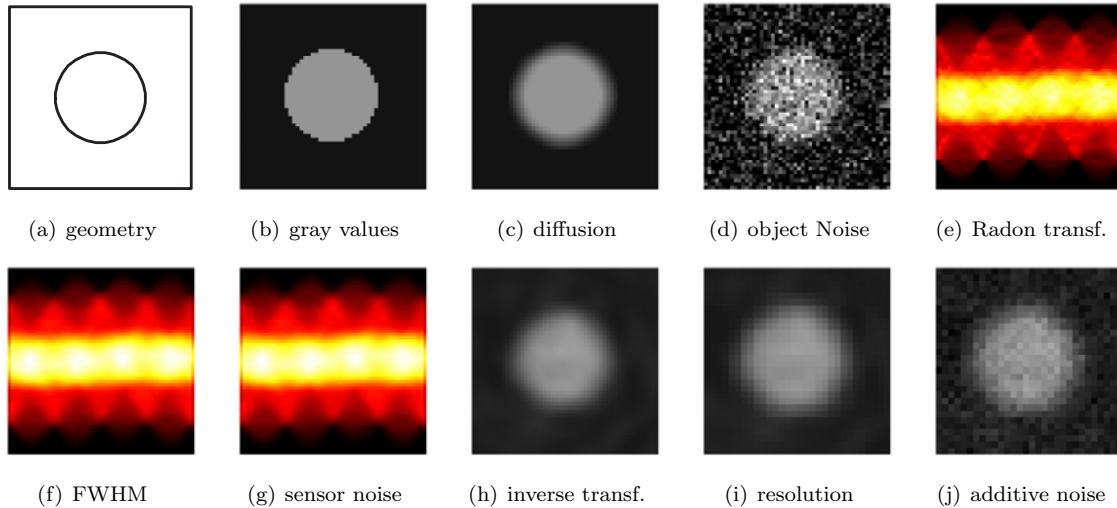


Figure 2: Simulation steps of software phantom (object diameter 25 mm; image size 50 mm × 50 mm; gray values 20/150; diffusion filter kernel size 5 pixel; Gaussian object noise $\mu = 0$, $\sigma = 30$; stepping angle for Radon transformation 3, 75°; FWHM = 2, 5 mm; Poisson sensor noise; inverse Radon transformation with linearer interpolation and Hamming filter; spatial resolution of 1,5 mm; Gaussian image noise $\mu = 0$, $\sigma = 5$).

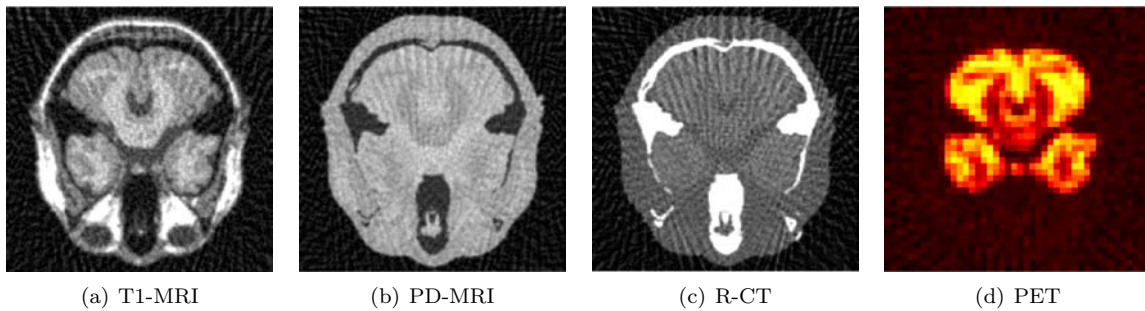


Figure 3: Simulated multimodal images of the brain.

3 Sample Simulations

In figure 3 some sample simulations for MRI, CT and PET are shown. To visualize the effect of the reconstruction artifacts these have been exaggerated compared to more realistic simulations. Visual evaluation by neuroradiologists showed that the simulated images are very realistic.

4 Simulation of Ultrasound Images

Since the principle of ultrasound imaging deviates significantly from the above modalities a separate module has been included in the software phantom.

This module is based on the ultrasound wave propagation software FIELD [6] and has been modified accordingly to simulate medical images.

Figure 4 shows some of the steps during the ultrasound simulation and the resulting gray value image.

Due to the complexity of the wave simulation the calculation time for the shown image on a 1.2 GHz Pentium III mobile (133 MHz front side bus, 1 GByte memory) accounted for 48 minutes. But this simulation allows the generation of very realistic image data while having control over all imaging parameters.

5 Numerical Quality Measures

To measure the accuracy of segmentation and registration algorithms several methods have been proposed in the literature (see [4] for an overview).

Apart from the above mentioned problem of reference data [7] one of the major difficulties is the application specific definition of quality or accuracy.

For some applications only the area or volume of certain objects might be of interest while for others the whole exact scene configuration is important.

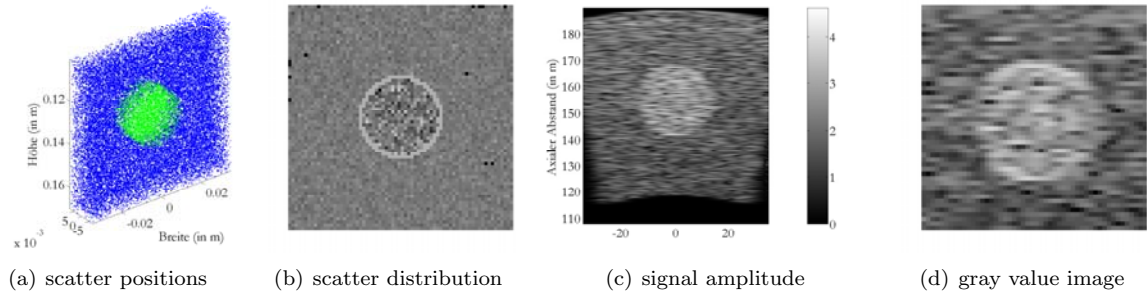


Figure 4: Steps of US simulation (parameters: 128 sensor elements; scan sector size 30° ; transducer frequency 6 MHz; 30 scan lines per element; 4 scatter point per pixel on average).

To account for these different aspects various numerical measures have been included in the proposed method. The user may then pick the one which is most relevant for the respective application.

5.1 Segmentation Quality

To estimate the quality of segmentation algorithms the following measures were included:

- absolute and relative number of misclassified voxels (area/volume size)
- Dice- and Tanimoto-coefficient (overlap of segments)
- distance of centres of gravity (location)
- eccentricity/solidity/compactness of objects (morphological measures)
- sensitivity/specificity (rel. number of false-right positive-negative segmented points)
- figure of merit (FOM: average normalized distance between the reference and the calculated segment borders; with a value of one indicating a perfect segmentation).

For a detailed explanation of the above measures and the respective equations see [4].

In figure 5 the output of the evaluation module for the FOM measure is shown for the results of a multimodal segmentation algorithm.

It can be seen clearly that the best results can be obtained for CT data and the largest error occurs for (low resolution) SPECT data. This indicates that the the model and/or parameters of the image processing algorithm are not as well adapted to the SPECT data as to the other imagery. Also it gives a rough estimate of the "real world" performance and a (relative) quantitative measure when comparing different algorithms and/or parameter settings.

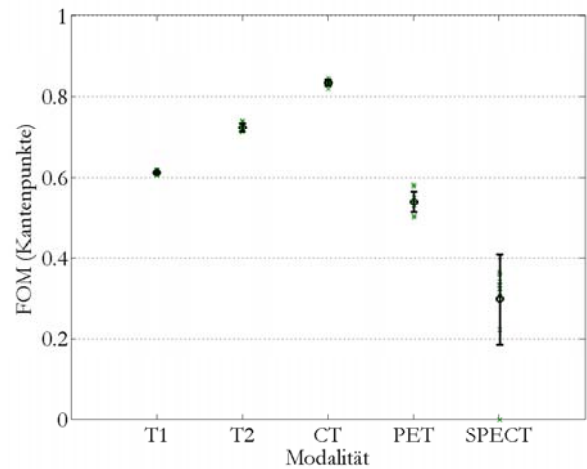


Figure 5: Display of automatically calculated results of FOM measure for a multimodal series of data with a segmentation algorithm (T1 and T2 are different MRI sequences, a value of FOM = 1 represents an ideal segmentation).

5.2 Registration Quality

Evaluation of registration accuracy is even more challenging than segmentation. Especially for elastic transformations deriving meaningful, i.e. intuitively understandable, numerical measures for differences between the calculated and the reference deformation fields is difficult because the transformation space is not (as much) restricted.

Affine transformations can be specified using translation, rotation, scale and shear but for elastic deformations the number of degrees of freedoms is much larger.

Especially in regions with low texture information there is usually a high registration error due to the high uncertainty in correlation. The importance of these regions is very application dependent – as is the regularization being performed in these areas.

Therefore registration error measures are usually calculated at relevant points/regions only. Two of the most widely used measures were implemented:

- vector difference (direction and absolute value as a graphical representation for every point and numerical measure with mean value and standard deviation for certain regions),
- fiducial registration error (FRE: normalized distance between calculated and reference location of selected points).

Specifically in medical image processing the fiducial registration error (FRE) gives an intuitive measure of the calculation error for selected target structures (fiducials) on the cost of global information about the calculated result.

6 Results

The developed tool allows to simulate various medical imaging modalities (MRI, CT, SPECT, PET, US) taking the most important parameters and artifacts into account. Parameterized series of data can be easily created and be used for automatic objective evaluation of segmentation and registration algorithms.

Numerous numerical quality measures have been included in the software to allow a task specific selection of the most relevant ones.

An automatically generated graphical presentation of the parameters and values of the resulting quality measures permits a very intuitive and fast evaluation.

This method and the respective software has already been used for development, optimization, evaluation and comparison of various segmentation and registration algorithms. It has proven to be a very valuable tool during these stages.

All simulation parameters can be saved for later use and documentation purposes which also allows to compare results between different research groups.

Due to the modular structure of the program it is easily extensible to include further modalities and parameters for a refined simulation and evaluation.

7 Discussion

One of the most critical points when creating reference data by simulation is the determination or at least estimation of how realistic they really are. This would allow a more precise estimation of "real world" results for a specific algorithm, rather than

the current relative performance measure within the "virtual world".

To our knowledge, so far no universal solution to this problem has been proposed in the literature. Therefore simulation should be considered as one – but very valuable – method amongst others to estimate the quality of image processing algorithms.

Also it should be noted that the general problem of contextual evaluation of segmentation and registration quality should be addressed in future versions as to consider application or even data dependent evaluation schemes.

One possibility to solve this problem would be to transform the data into a standardized anatomical coordinate system and define task specific target regions or structures. These important regions will then be assigned a higher weight compared to other structures when calculating the average (global) quality measure.

8 Acknowledgements

I would like to thank the German foundation "Studienstiftung des Deutschen Volkes" for providing the financial support of this project and R. Freyer, U. Morgenstern and my other colleagues for their help and advices.

References

- [1] I. C. Research Systems, "Visible human CD - user's guide version 1.0." CD, E- Mail: visible@rsinc.com, 1995.
- [2] D. Collins, A. Zijdenbos, V. Kollokian, J. Sled, N. Kabani, C. Holmes, and A. Evans, "Design and construction of a realistic digital brain phantom," *IEEE Transactions on Medical Imaging*, vol. 17, pp. 463–468, Mar 1998.
- [3] C. Cocosco, V. Kollokian, R.-S. Kwan, and A. Evans, "BrainWeb: Online interface to a 3D MRI simulated brain database," *NeuroImage*, vol. 5, p. 425, May 1997.
- [4] F. Uhlemann, *Wahrscheinlichkeitsbasiertes Modell zur verknüpften Segmentierungs- und elastischen Deformationsanalyse in der medizinischen Bildverarbeitung*. PhD thesis, Technische Universität Dresden, Fakultät Elektrotechnik und Informationstechnik, March 2006.
- [5] A. K. Jain, *Fundamentals of digital image processing*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989.

- [6] J. Jensen, "Field: A Program for Simulating Ultrasound Systems," in *10th Nordic-Baltic Conference on Biomedical Imaging Published in Medical & Biological Engineering & Computing*, vol. 34, Suppl. 1., (Tampere), pp. 351–353, Jun 1996.
- [7] T. M. Lehmann, "From plastic to gold: A unified classification scheme for reference standards in medical image processing," *SPIE*, vol. 4684, pp. 212–231, 2002.

Analysis of Differential Interference Contrast Microscopy Images of the Retina

D. H. Wojtas¹, B. Wu¹, P. Wenig¹, P. K. Ahnelt², P. J. Bones¹ and R. P. Millane¹

¹Computational Imaging Group, Department of Electrical and Computer Engineering
University of Canterbury, Private Bag 4800, Christchurch, New Zealand

²Department of Physiology, Vienna Medical University, Schwarzschanerstr 17, A-1090 Vienna, Austria
Email: dhw32@student.canterbury.ac.nz

Abstract

An algorithm is presented for processing and analysis of differential interference contrast (DIC) microscopy images of the retina to study the cone mosaic. The algorithm utilizes a number of characteristics of the DIC retinal image to locate the cones to a high degree of accuracy. This information is used to analyse the cone size distribution, the spatial distribution of cone density, and short-range order and domain structure of the mosaic.

Keywords: retina, fovea, cone, differential interference contrast microscopy, image analysis

1 Introduction

The retina is a thin layer of neural cells that lines the inner projection plane of the vertebrate eye. Within the retina is a sub-layer of photoreceptor cells (rods and cones) that are responsible for the processing of light via patterned excitation. Positioned where the optical axis reaches the back of the eye is a specialized region of the retina termed the macula. At the centre of the macula, primates possess a single, concentrated region of photoreceptors known as the fovea. The foveal mosaic is comprised mostly of cones; photoreceptors that function only in relatively bright light. The fovea is responsible for our sharp central vision [1].

Analysis of the spatial pattern of photoreceptors in the retina has both medical and academic research interests. It is used to determine the regularity of a particular retinal region of an animal species during its development [2]. Spatial regularity studies contribute to determining a unifying mechanism that could underlie all mosaic patterns, and to detecting degradation of mosaic quality from pathologic alterations including photoreceptor loss.

Previous studies on the foveal photoreceptor topography have primarily used images by light microscopy of histologic sections [3], and recently, images have been obtained of *in vivo* samples by adaptive optics ophthalmoscopy [4]. Differential interference contrast (DIC) microscopy is an optical microscopy illumination technique with the potential ability of visualising finer details of the photoreceptor topography in isolated intact retina. The method converts the gradients in optical path length into amplitude differences

in the image by use of Wollaston prisms, and imposes an apparent light direction on the formed image [5]. DIC microscopy is an excellent technique for obtaining optical cross sections of unstained retinal sub-layers as it produces very high contrast at the edges of biological structures. DIC image resolution and clarity are unrivaled among standard optical microscopy techniques. However, the apparent light source direction poses a challenge for digital analysis of DIC images. Previous studies using DIC images of the foveal photoreceptor mosaic have used manual analysis techniques [6],[7].

In Section 2, an algorithm for location of the cone boundaries and centres is described, and the accuracy assessed. In Section 3, results from structural analysis of the foveal mosaic are presented. Concluding remarks are made in Section 4.

2 Cone location algorithm

2.1 Preprocessing

Figure 1 is a DIC image that shows an optical cross-section of a rod-free foveal region of a human retina. The image is 868×606 pixels and represents a $170\mu m \times 120\mu m$ region of the retina. The origin is defined to be at the bottom-left corner of the image. From Figure 1 one notices an apparent light source direction of 135° .

Since the DIC microscopy image represents the gradient of the image intensity profile along the shear axis [5], the original intensity image can, in principle, be obtained by integrating the DIC image along this direction. Although we have had

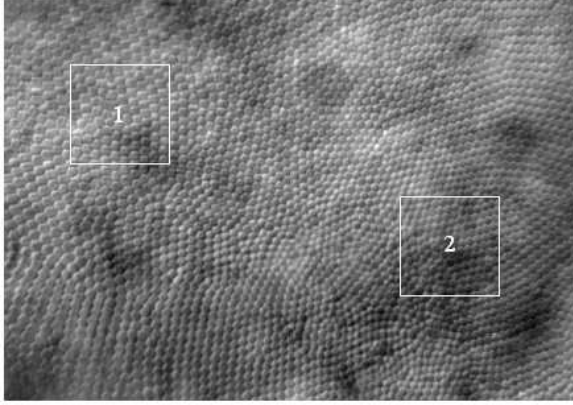


Figure 1: DIC image of the human cone mosaic. Regions 1 & 2 are shown enlarged in Figure 3.

some success using leaky integration in images with well separated cones, integration has not proved effective for the tightly packed cones in the macula region. Therefore, the following algorithm was developed to locate the cone centres directly from the DIC image.

The original image suffers from background variation. Median filtering, with a window of dimensions 1.5 times the largest cone diameter, was used to estimate the background. This was then subtracted from the original image.

2.2 Cone centres

Nearly every cone in the mosaic exhibits a regional maximum in intensity. Regional maxima of the image were isolated by performing extended maxima extraction (EME) [8], which produces a binary image consisting of just the maximal features. Missed or multiple maxima were manually corrected. In total, $\sim 3\%$ of the cones required manual location of the maxima. The centroids, m_i , of the features were then obtained so that the N cones in the image could be indexed by its regional maximum centroid position (\bullet in Figure 2a). Although the maxima index the cones in a DIC image they are not at the centres of the cones. The next step is therefore to determine the cone centres from the maxima.

The mean nearest-neighbour distance, $\bar{d}_{NN,i}$, was determined for each cone maximum. Nearest-neighbour maxima were determined using Voronoi neighbour identification. For a cone i ,

$$\bar{d}_{NN,i} = \left(\frac{1}{n}\right) \sum_{j=1}^n \text{dis}(m_i, m_j), \quad (1)$$

where n is the number of detected neighbours and $\text{dis}(m_i, m_j)$ is the distance to neighbouring maxima m_j . The $\bar{d}_{NN,i}$ values were useful for subsequent image analysis.

For each cone, a point was defined a distance $\frac{1}{4}\bar{d}_{NN,i}$ from the detected maximum centroid position in the direction opposite to the apparent light source direction (+ in Figure 2a). This set of points is a good preliminary approximation of the cone centres. However, closer inspection showed that only $\sim 15\%$ of the cone centres were identified to within 1 pixel of manually detected centres.

2.3 Cone boundaries

The centre positions were improved by first estimating the cone boundaries. Inspection of the image shows that a good estimate of the cone boundary can be obtained by locating the dark edges that surround the maxima. The dark edges were first highlighted by performing median filtering with a 2×2 window, and contrast enhancement by histogram equalization. The image was then segmented into rectangular regions, each containing ~ 100 cone maximums. This was a crucial step so that subsequent processing could accommodate the gradual shift in cone density across the mosaic. A regional average of the $\bar{d}_{NN,i}$ values, $\bar{d}_{NN,r}$, was calculated from the cones present in a given region. This number was used as a measure of the average cone size in a region.

In each region, the image was thresholded with a cutoff equal to the lower quartile of the pixel intensities within the region. Residual small features with fewer than $A_{opt,r}$ edge connected pixels were removed, leaving a binary image of just the dark edges of the cones (Figure 2b). The optimal area for a region was determined to be $A_{opt,r} = (1/10)\pi(\bar{d}_{NN,r}/2)^2$.

For each cone, N_R radial lines with constant angular separation originating from the approximate cone centre were constructed (Figure 2b). We used $N_R = 48$. The first zero pixel along each radial line from the approximate centre was taken as a possible boundary marker (\square in figure 2b). The distance of the marker from the centre was constrained to lie in the interval $(0.25\bar{d}_{NN,i}, \bar{d}_{NN,i})$.

The boundary markers for a cone were then assigned to groups according to their contiguity. Two criteria were used to define a contiguous group; each identified boundary marker in the group had to lie on neighbouring radial line segments, and each identified boundary marker had to be at a radial distance within $(0.1)\bar{d}_{NN,i}$ of the radial distance of the preceding boundary marker in that group.

The group size, S_g , and the number of radial lines to adjacent boundary markers anticlockwise and

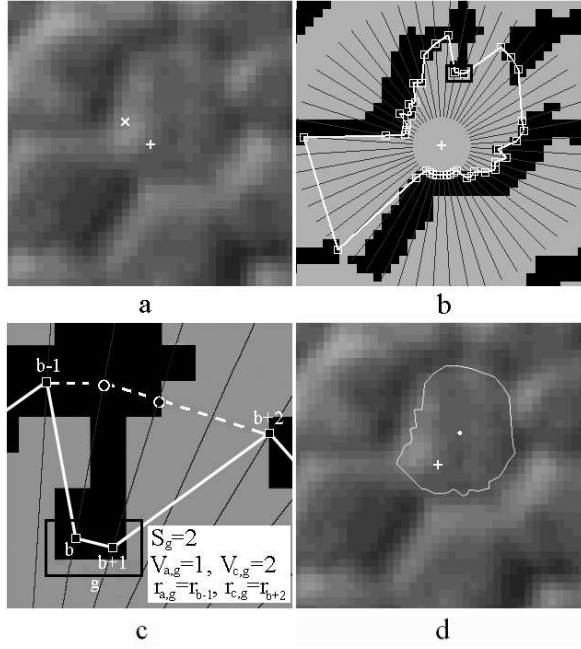


Figure 2: **a)** A region of the image showing a cone with its maximum position (\times) and approximated cone centre ($+$). **b)** The extracted dark edges used in determining a cone boundary. The initial estimate of the cone boundary is marked with \square s. **c)** A close-up of the group of boundary markers in the box of Figure 2b. The readjusted marker positions are shown (\circ). **d)** The final estimate of the cone boundary with its centroid (\bullet).

clockwise to a group (denoted $V_{a,g}$ and $V_{c,g}$ respectively) were determined for each group. These parameters were used to ‘smooth’ the initial estimate of the boundary. If $S_g \leq (0.05)N_R$, the boundary markers, b , in the group have their radial distances redefined as

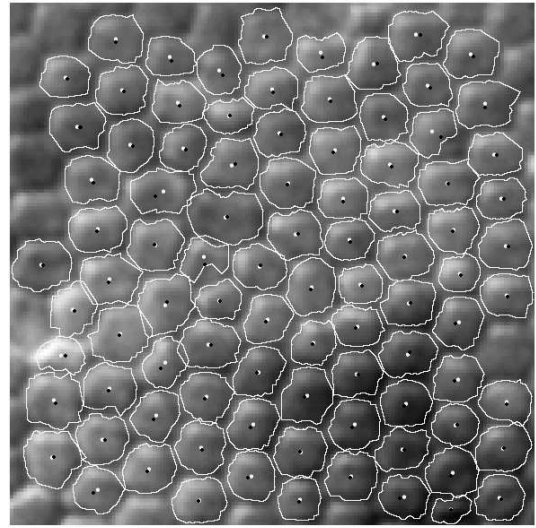
$$r_b = (V_{a,g}r_{a,g} + V_{c,g}r_{c,g}) / (V_{a,g} + V_{c,g}) \quad (2)$$

where $r_{c,g}$ and $r_{a,g}$ are the distances of the adjacent boundary markers clockwise and anticlockwise to the group. Figure 2c illustrates this procedure for the group of boundary markers enclosed by the rectangular box in Figure 2b. The circles represent the redefined positions calculated using Equation 2.

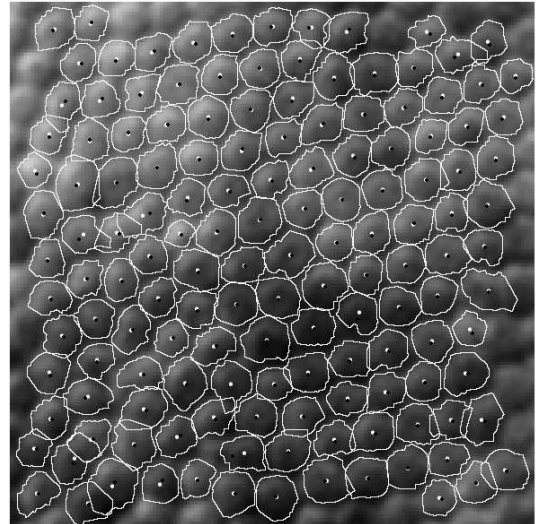
The $V_{a,g}$ and $V_{c,g}$ values in Equation 2 remain constant whereas $r_{a,g}$ and $r_{c,g}$ may change. The smoothing routine is performed repeatedly until no readjustment is performed. The resulting set of N_b detected boundary markers provides an accurate mapping of the cone boundary. A polygon of N_b sides is constructed from this set and the centroid of the polygon determined, giving the final estimate of the cone centre (Figure 2d).

2.4 Accuracy of the algorithm

To determine the accuracy of the algorithm, two regions of the mosaic were selected (Figure 1), in which the cone centres were determined both automatically and manually. The first region contains 95 cones, and the mosaic appears almost crystalline. The second region contains 144 cones, and includes part of the region where the mosaic appears somewhat amorphous. 93% of the cones centres in region 1 and 95% in region 2 were within one pixel of the manually determined positions (Figure 3). This is considered a good result for this image.



Region 1



Region 2

Figure 3: Comparison between manually (black dots) and automatically (white dots) determined cone centres. Estimated cone boundaries are also shown.

3 Analysis of the mosaic

3.1 Cone size and packing density

An advantage of the algorithm described above is that it gives information on cone size and shape. The cross-sectional area, A_i , of a cone can be estimated as the area enclosed by the polygon tracing the cone boundary, and the cone size can be estimated as the diameter of a circular disc of equivalent area. Figure 4 shows the histogram of cone diameters. The average diameter was found to be $2.5\mu\text{m}$.

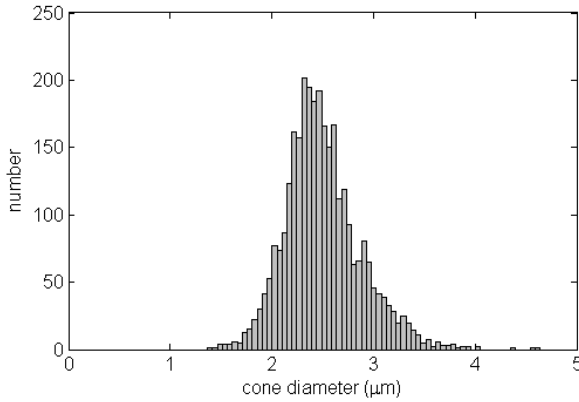


Figure 4: Histogram of cone diameters.

By determining the area occupied by cones in the mosaic, one can determine the packing density, $\rho = \sum_i A_i/A$, where A is the area of the image. A perfect hexagonal arrangement of circular cells would correspond to $\rho = 0.91$. For this particular retinal image, $\rho = 0.77$. The smaller value obtained can be attributed to variations in the cone size and imperfect hexagonal packing.

3.2 Cone nearest-neighbours

Identification of nearest-neighbour cones is necessary for further analysis of the mosaic. Nearest-neighbour cones are cones that have direct contact with each other, and are clearly discernable to the human eye. Determining the neighbourhood of each cone was conducted in two steps.

First, the Voronoi neighbours for each cone were identified. Due to occasional irregular arrangements in the cone positions, there are cases where genuine neighbours are not Voronoi neighbours. Lattice lines, the connecting lines between the centres of two nearest neighbours, give an indication of the orientation and spatial arrangement of the cones. If nearest neighbours have been correctly identified and the underlying array is sufficiently regular, one expects the lattice lines to form a triangular ‘net’.

An additional check for nearest neighbours was performed based upon the underlying mosaic topology. In areas where Voronoi detection was unsuccessful, polygon shaped ‘holes’ in the net are present (Figure 5a). The ratio of the inter-cone distance to the average cone-to-Voronoi-neighbour distance was used as a metric to decide which pair of cones at the corners of these ‘holes’ are most likely to be neighbours. Referring to Figure 5a, if

$$\frac{dis(B,C)}{\rho(B)} + \frac{dis(B,C)}{\rho(C)} < \frac{dis(A,D)}{\rho(A)} + \frac{dis(A,D)}{\rho(D)} \quad (3)$$

where $\rho(X) = \frac{1}{5} \sum_i dis(X, Neighbour_i)$, then points B and C are identified as nearest neighbours (Figure 5b).

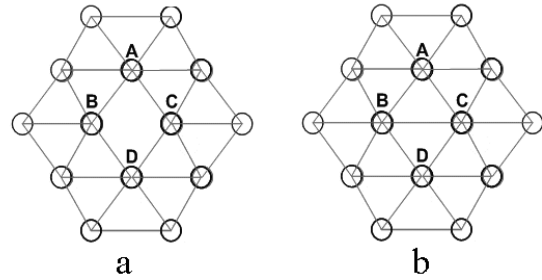


Figure 5: a) The neighbours associated using Voronoi identification. b) By comparing the metrics, cones B-C are more likely to be neighbours.

3.3 Spatial autocorrelation

The autocorrelogram can be used to visualise spatial trends in neighbour relationships and aspects of mosaic topography. To produce an autocorrelogram, the position of each cone is sequentially made the central reference point and relative positions of all other cones are plotted. In effect, a unique mosaic for each cone is derived. The autocorrelogram is the superposition all such mosaics.

Since the orientation of the lattice net is not constant across the image, a direct calculation will ‘wash-out’ angular structure in the autocorrelogram. Therefore, the local orientation of the hexagon formed by the neighbours of cones with six neighbours (referred to as 6n-cones) is subtracted out before the autocorrelogram is calculated. The autocorrelogram calculated is shown in Figure 6 out to a distance of $20\mu\text{m}$. The prominence of hexagonal packing is clearly visible.

3.4 The foveal centre

Figure 7 shows a surface plot of the nearest neighbour distance in each Voronoi domain after being passed through a median filter. The nearest

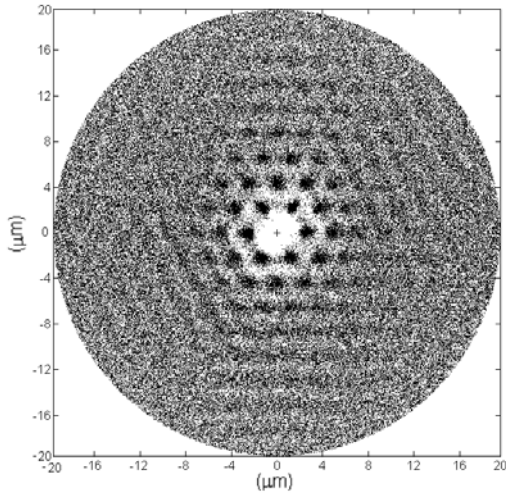


Figure 6: The spatial autocorrelogram obtained by superimposing the relative neighbourhoods of each cone.

neighbour distance is taken as an estimate of cone diameter. It is clear from Figure 7 that a region containing the smallest cone sizes exists. This region is termed the foveola of the retina, the rod-free centre of the fovea. A point of smallest cone size was approximated by averaging the positions of the 50 cones with smallest $\bar{d}_{NN,i}$ values (the white point below-right of Figure 7).

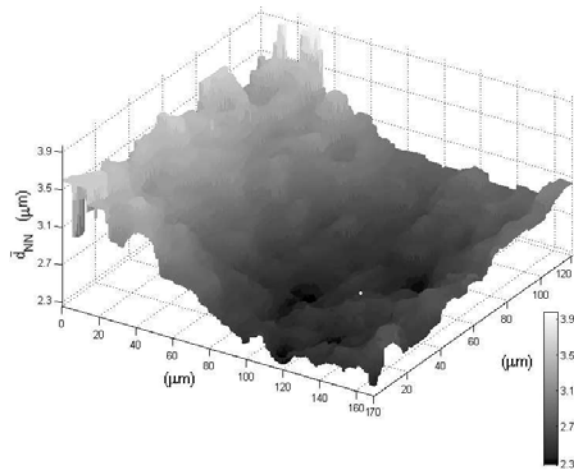


Figure 7: A surface plot mapping the average cone diameter. The white point indicates the position of the foveal centre.

A useful technique used by Pum et.al. [3] to extract information regarding the topographic properties in the foveal region involves calculating the so-called “regularity ratio”. The regularity ratio is the number of cones with 6 neighbours (6n-type cones) divided by the sum of the number cones with 5 neighbours (5n-type cones) and the number of cones with 7 neighbours (7n-type cones). This is calculated within concentric rings of increasing distance from the foveal centre. The regularity

ratio is plotted in Figure 8 as a function of this distance.

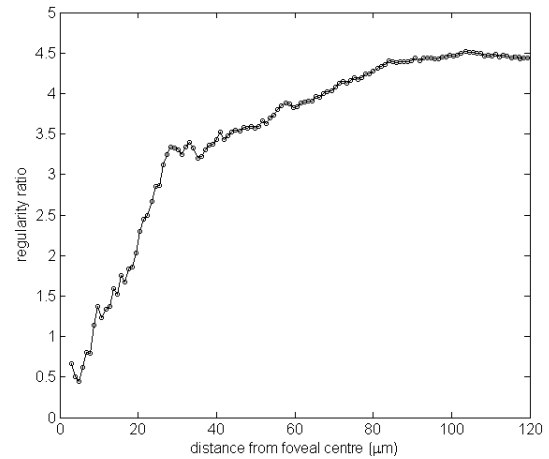


Figure 8: The regularity ratio versus distance from the foveal centre.

The distance at which the regularity ratio begins to level off signifies the onset of crystalline order. From Figure 8, this occurs at a distance of $\sim 30\mu m$. In the crystalline regions cones are organized hexagonally, and so most cones are of 6n-type. The low 6n-type cone population within the foveola region is a result of the amorphous mosaic organization, i.e. a lack of hexagonal organization. The 5n/7n cones are considered to be a direct indication of non-crystalline topology. Lattice order appears to be at a maximum in the regions surrounding foveal centre, as reported in [3].

3.5 Domain identification

Abrupt rotational shifts of the orientations of the net are discernable in Figure 1. These shifts appear to mark the boundaries between ordered mosaic domains. Further investigation shows that 5n/7n cone pairings are common along such boundaries [3]. We developed an algorithm to segregate these ordered domains and their boundaries in the mosaic.

A domain was defined as a group of neighbouring 6n cones which have an axial orientation within 4° of at least one of its nearest neighbour’s orientation. This value for a maximum neighbour orientational shift was an arbitrary choice. Generally, the rotational shift in axial orientation experienced at a domain boundary is larger than this ($\sim 10^\circ$ [3]). A minimum domain size of 3 cones was used. Domains consisting of one or two cones were considered to be part of an amorphous topography.

Figure 9 shows the domain structure of the retinal mosaic. 5n, 7n and 8n cones as well as 6n cones

which are not included in a domain are indicated by white lattice lines. Each crystalline domain of 6n cones has black lattice lines. Lattice lines connecting neighbouring cones positioned either side of a domain interface are coloured white. Lattice lines for cones with less than 5 neighbours are not plotted as they generally lie around the edge of the image. A prominent region of amorphous topography is observed in the vicinity of the identified foveal centre.

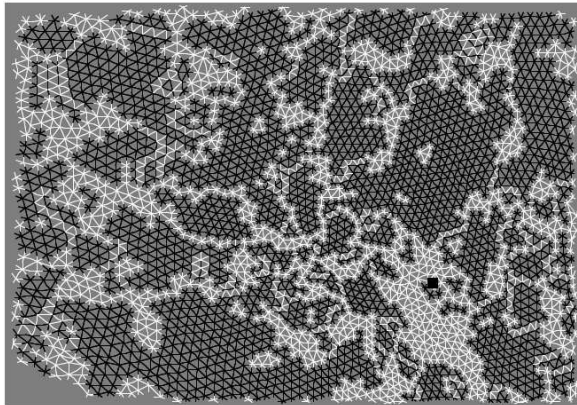


Figure 9: The domain structure of the retinal mosaic. The black square indicates the position of the foveal centre.

4 Conclusions

An algorithm has been developed for automatically determining the boundaries and locations of cones in an image of a portion of the foveal retina obtained by differential interference contrast (DIC) microscopy. This particular kind of image required a somewhat involved algorithm but requires minimal user input and performs to an accuracy of $\sim 94\%$ compared to a manual analysis. More work is required to test the algorithm on other retinal DIC images in order to determine the versatility of the methodology described here.

The cone centres obtained are useful for analysing the mosaic structure in terms of packing density, the distribution of cone size, short-range and long-range order, and determination of ordered regions. There is further potential for detailed analysis of the disorder in the retina and other mosaic arrays.

5 Acknowledgements

We are grateful to the N.Z. Marsden Fund for financial support.

References

- [1] P. K. Ahnelt and H. Kolb, "The mammalian photoreceptor mosaic-adaptive design," *Progress in Retinal and Eye Research*, vol. 19, no. 6, pp. 711–777, 2000.
- [2] J. E. Cook, "Spatial regularity among retinal neurons," in *The Visual Neurosciences* (L. M. Chalupa and J. S. Warner, eds.), pp. 485–495, MIT: A Bradford Book, 2004.
- [3] D. Pum, P. K. Ahnelt, and M. Grasl, "Iso-orientation areas in the foveal cone mosaic," *Vis. Neuro.*, vol. 5, no. 6, pp. 511–523, 1990.
- [4] A. Roorda, "Adaptive optics ophthalmoscopy," *Journal of Refractive Surgery*, vol. 16, no. 5, pp. 602–607, 2000.
- [5] D. Murphy, "Differential interference contrast microscopy and modulation contrast microscopy," in *Fundamentals of Light Microscopy and Digital Imaging* (L. M. Chalupa and J. S. Warner, eds.), pp. 153–168, New York: Wiley, 2001.
- [6] C. A. Curcio, K. R. Sloan, R. E. Kalina, and A. E. Hendrickson, "Human photoreceptor topography," *Journal of Comparative Neurology*, vol. 292, no. 4, pp. 497–523, 1990.
- [7] O. Packer, A. E. Hendrickson, and C. A. Curcio, "Photoreceptor topography of the retina in the adult pigtail macaque (*Macaca nemestrina*)," *Journal of Comparative Neurology*, vol. 288, no. 1, pp. 165–183, 1989.
- [8] L. Vincent, "Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms," in *IEEE, vol. 2 of Trans. on Image Process*, pp. 176–201, 1993.

ROBPCA-SIFT: a feature point extraction method for the consistent with epipolar geometry in endoscopic images

J.S. Oh¹, H.C. Kim^{2,3}, J.M. Koo¹, J.S. Yu¹, T.H. Kang¹, J.D. Lee⁴, and M.G. Kim^{1*}

¹ Korea Univ., Department Electronics and Information Engineering, Seoul, Korea.

² Korea Univ., Biomedical Engineering, Biomedical Science of brain Korea 21, Seoul, Korea.

³ Korea Univ., Korea Artificial Organ Center, Seoul, Korea.

⁴ Konkuk Univ., Department of Gastroenterology/Hepatology, College of Medicine, Chungju, Korea.

Email: mgkim@korea.ac.kr

Abstract

The researches that use geometrical information in computer vision are very actively developing field. One of the main problems in the multi-view geometry is a method of feature extraction to find corresponding points between successive frames. Up to now a feature point extraction which uses SIFT (Scale Invariant Feature Transform) is excellent among the algorithm to be known. SIFT parameters that have been used in general image however, did not have good performance for endoscopic images. Furthermore, we used ROBPCA (Robust Principal Component Analysis) to find rather good feature points in endoscopic images with the noise. ROBPCA-SIFT draws salient feature points from endoscopic images to find the probe postures and reconstruct 3D structures. The purpose of this study is to extract 3D information by using prevailing endoscope as it is without any modification (i.e., not by using so called stereo endoscope).

Keywords: feature extraction, SIFT, ROBPCA, endoscopic image, fundamental matrix.

1 Introduction

Inferring 3D structure information from a sequence of endoscopic images is important for CAD (Computer Aided Diagnostic) system. For this purpose, three steps of processing are performed. First, feature extraction is needed for matching problem. Matching problem is worked out by clustering. Second, calculate the fundamental matrix using the well corresponding points. Finally, the 3D projective reconstruction is computed from the fundamental matrix. The main goal of this paper is extraction of accurate and reliable feature points on noised endoscopic images. Search for Harris affine invariant point detector [1] and SIFT (Scale Invariant Feature Transform) [2]. These methods are reliable in general images.

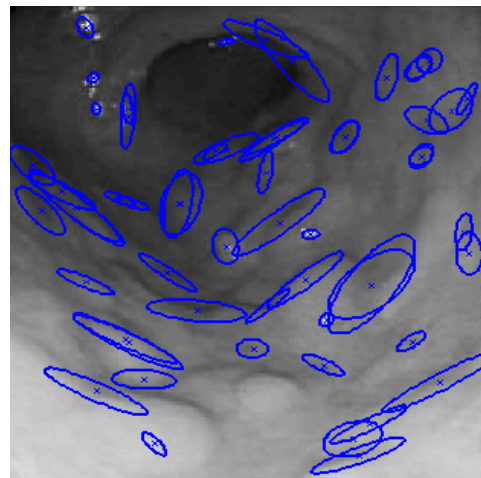
2 Extraction Feature Points in Endoscopic Image

Feature points should be extracted for well matching between images. Two major approaches for feature extraction for natural views are area based methods and feature base methods. These methods have both advantages and disadvantages depend on application. Unfortunately, both methods are not relevant for the endoscopic images, because the scales of target object change between images. An hybrid method called SIFT (Scale Invariant Feature Transform) was

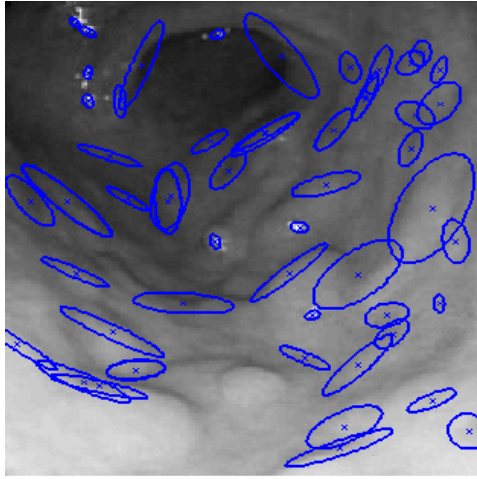
required to extract scale invariant or close to affine invariant features. SIFT method showed excellent performance when compared with alternative methods (Harris Affine invariant point detector etc.) for general images.

2.1 Harris Affine Invariant Point Detector

Consider Harris affine invariant point detector for extraction feature points in endoscopic images. This method is described in [1]. The method operates well in general images, but not appropriate method for endoscopic images.



(a)



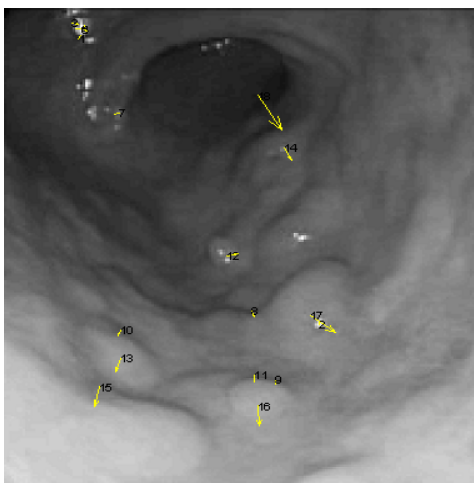
(b)

Figure 1: Feature points using Harris affine invariant point detector in endoscopic images. (a) first image. (b) second image.

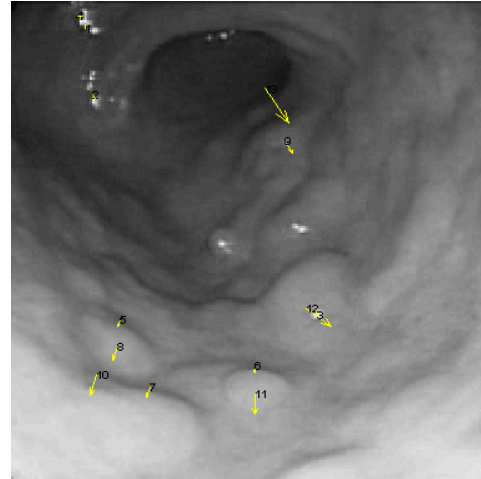
The reason can be explained as follow. We got 61 feature points in Fig.1 (a) and 57 feature points in Fig.1 (b). As we can see clearly in Fig.1, the possible matching points are not enough sufficient in order to extract some postures information between the two images. Also, we have another problem that feature points are not exact, because the lighting conditions when take picture may different. These phenomena give serious influence when we solve the matching problem. Consequently the method is not appropriate and we may necessary some scale invariant techniques.

2.2 Scale Invariant Feature Transform (SIFT)

SIFT, as described in [2]. This method operates well like the Harris affine invariant point detector in general images. However, the algorithm with parameter (contrast threshold) value to be induced does not operate well for endoscopic images.



(a)

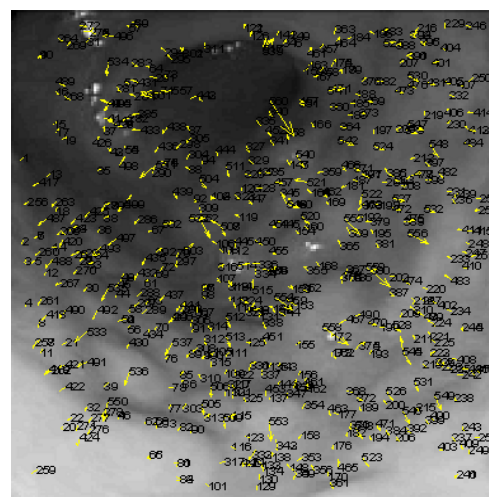


(b)

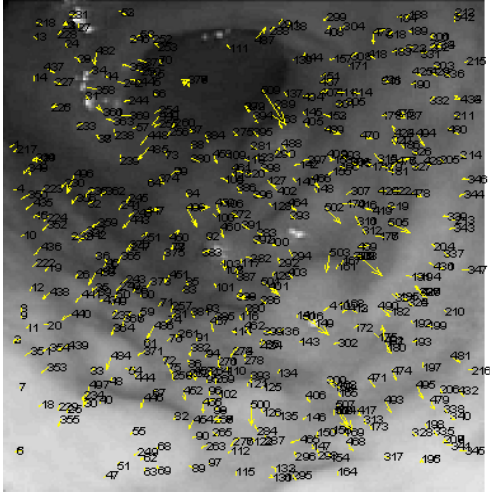
Figure 2: Feature points using SIFT in endoscopic images. (a) first image. (b) second image.

We got 18 feature points with a contrast threshold of 0.02 in Fig.1 (a) and 13 feature points with a contrast threshold of 0.02 in Fig.1 (b). So we may need to change the value. To change the parameter we look into the algorithm to modify a parameter value. SIFT consists of four major stages: (1) scale-space extrema detection; (2) keypoint localization; (3) orientation assignment; (4) keypoint descriptor. A contrast threshold among the parameters is the second stage among these stages. The function value at the extremum, $D(\hat{X})$, is useful for rejecting unstable extrema with low contrast. Definition of $D(\hat{X})$ is showed in the equation (1). D is the difference-of-Gaussian function. We use the histogram of the $|D(\hat{X})|$ to get sufficient feature points.

$$D(\hat{X}) = D + \frac{1}{2} \frac{\partial D^T}{\partial X} \hat{X} \quad (1)$$



(a)



(b)

Figure 3: Feature points using SIFT with changed parameter value in endoscopic images. (a) first image. (b) second image.

Then, we got 560 feature points with a contrast threshold of 0.0031 in Fig.1 (a) and 509 feature points with a contrast threshold of 0.0032 in Fig.1 (b). The goal of this section is extraction of feature points that controlled by contrast threshold among the parameters for applying SIFT method in endoscopic images. Experimental study shows that resulting feature points for endoscopic images can have good distribution and meaningful control number of feature points than traditional methods. The good distribution of the feature points result in good characteristic of the fundamental matrix.

3 CLAPCA-SIFT and ROBPCA-SIFT

The purpose to use classical PCA or ROBPCA is reduction of dimensions of the descriptor that is described in [2].

3.1 CLAPCA-SIFT

PCA-SIFT, as described in [5]. However, this method does not operate well in our application. It is because the gradient values of local patches of our images does not change much, this means we can not distinguish the patches.

Let's call CLAPCA-SIFT to distinguish with PCA-SIFT. This method does not use the descriptor described in [5]. CLAPCA-SIFT combines classical PCA and the descriptor described in [2]. This method is used to reduce 128 elements to small number. We use 20 components or 10 components in experiments with endoscopic images.

3.2 ROBPCA-SIFT

ROBPCA, as described in [7]. This algorithm combines projection pursuit in high dimensions and

MCD (Minimum Covariance Determinant) in low dimensions. ROBPCA consists of three major stages:

- Stage 1. We start by reducing the data space to the affine subspace spanned by the n observations.
- Stage 2. We try to find the $h < n$ 'least outlying' data points.
- Stage 3. We robustly estimate the scatter matrix of the data points in X_{n,k_0}^* using the MCD estimator.

$$X_{n,k_0}^* = (X_{n,r_1} - \mathbf{1}_n \hat{\mu}'_1) P_{r_1, k_0} \quad (2)$$

This method offers robust principal components. ROBPCA-SIFT combines ROBPCA and the descriptor described in [2] as CLAPCA-SIFT.

4 Fundamental Matrix and 3D Projective Reconstruction

We need to calculate the fundamental matrix for the 3D projective reconstruction. From the fundamental matrix some very useful geometric information can be extracted. For example, the postures (i.e., rotation matrices and translation vectors) of the sequence of endoscopic images. The fundamental matrix satisfies the condition that for any fair of corresponding points $x \leftrightarrow x'$ in the two images.

$$x'^T F x = 0 \quad (3)$$

The fundamental matrix is that we found to minimize the geometric errors. The geometric error (d, d') is showed in Fig.4.

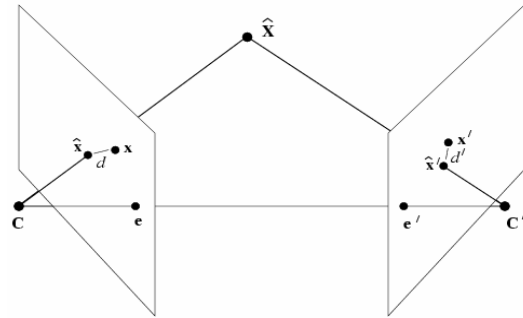


Figure 4: Minimization of geometric error (d, d').

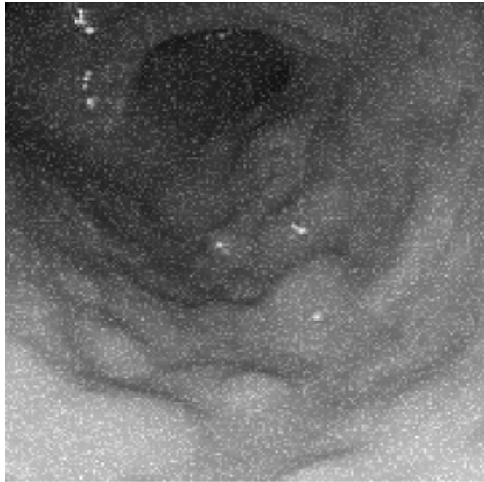
We seek the points \hat{x} and \hat{x}' that minimize the function

$$C(x, x') = d(x, \hat{x})^2 + d(x', \hat{x}')^2 \text{ subject to } x'^T F x = 0 \quad (4)$$

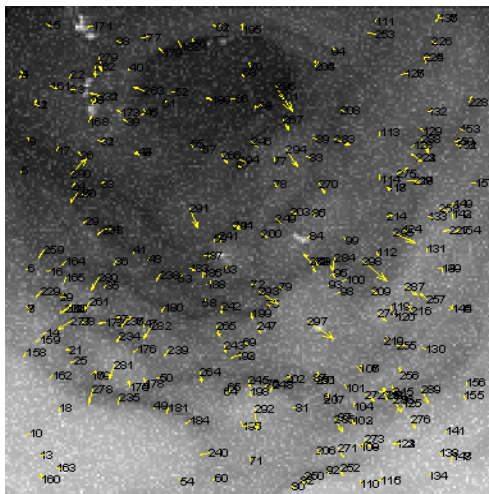
where $d(\bullet, \bullet)$ is the Euclidean distance between the points. The camera matrices for each of endoscopic image sequences are computed from the fundamental matrix. And then the 3D structure \hat{x} is obtained by a triangulation method. The fundamental matrix and 3D projective reconstruction are explained in detail in [9].

5 Experiments and Results

We use Gaussian noise (mean: 0, variance: 0.005) for a performance comparison. We added the noise to second image and used at the experiment.



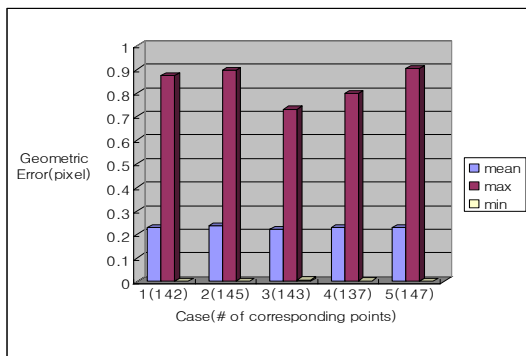
(a)



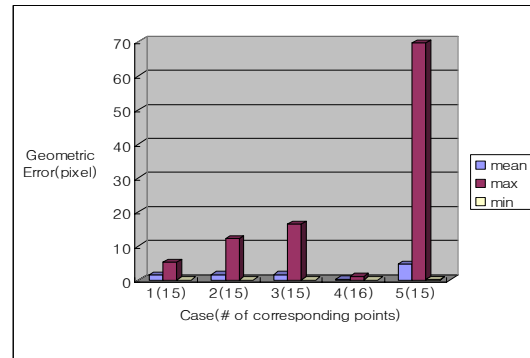
(b)

Figure 5: The endoscopic second image. (a) added the Gaussian noise. (b) feature points of (a).

We got 298 feature points with a contrast threshold of 0.0065 in Fig.5 (b).



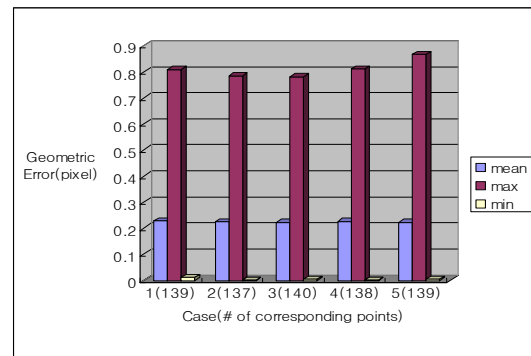
(a)



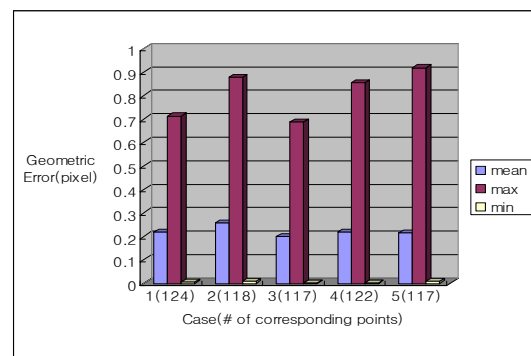
(b)

Figure 6: Geometric errors using SIFT in endoscopic image. (a) without the noise. (b) with the noise.

We got about 140 corresponding points in endoscopic image without the noise and about 15 corresponding points in endoscopic image with the noise. SIFT has the accuracy of half pixel. This value is becomes the criteria to evaluate the performance of the algorithm. Fig.6 shows that SIFT method operates well in endoscopic image without the noise, but does not operates well in endoscopic image with noise. Because mean of geometric errors is smaller than 0.5pixel in Fig.6 (a) but mean of geometric errors is bigger than 0.5pixel in Fig.6 (b). Specially, max of geometric errors has very big value in Fig.6 (b).



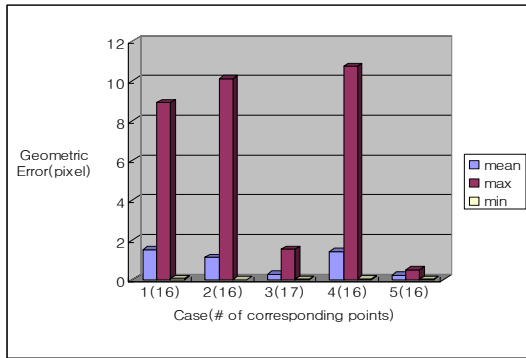
(a) CLAPCA-SIFT20



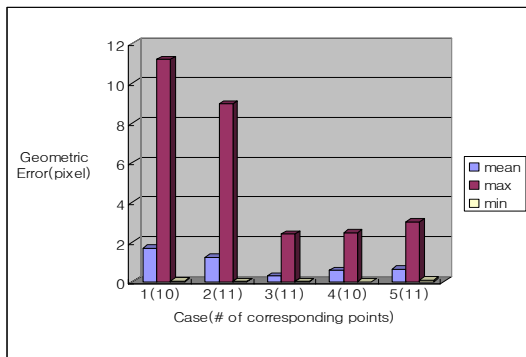
(b) CLAPCA-SIFT10

Figure 7: Geometric errors in endoscopic image without the noise. (a) using CLAPCA-SIFT (with 20 components). (b) using CLAPCA-SIFT (with 10 components).

Mean of geometric errors are smaller than 0.5pixel and max of geometric errors is small in Fig.7. Both CLAPCA-SIFT20 and CLAPCA-SIFT10 operate well such as SIFT in endoscopic image without the noise.



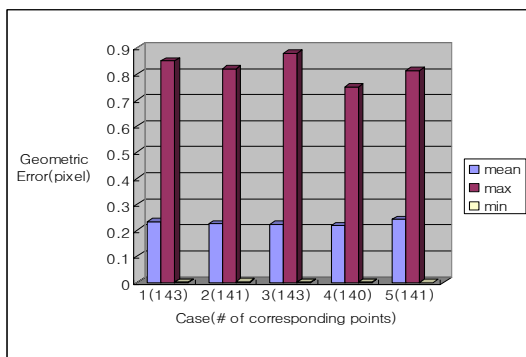
(a) CLAPCA-SIFT20



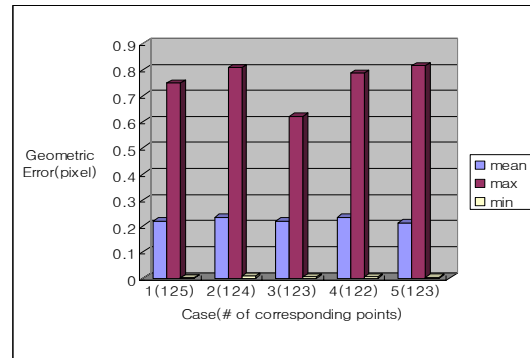
(b) CLAPCA-SIFT10

Figure 8: Geometric errors in endoscopic image with the noise. (a) using CLAPCA-SIFT (with 20 components). (b) using CLAPCA-SIFT (with 10 components).

However, mean of geometric errors is bigger than 0.5pixel and max of geometric errors is big in Fig.8. CLAPCA-SIFT20 and CLAPCA-SIFT10 does not operate well in endoscopic image with the noise. CLAPCA-SIFT is unreliable in endoscopic image with the noise.



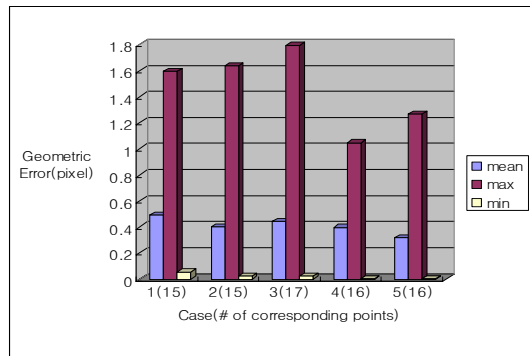
(a) ROBPCA-SIFT20



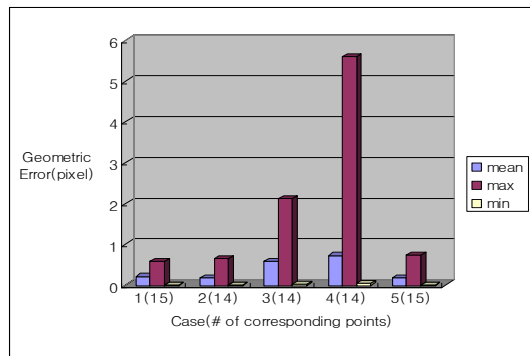
(b) ROBPCA-SIFT10

Figure 9: Geometric errors in endoscopic image without the noise. (a) using ROBPCA-SIFT (with 20 components). (b) using ROBPCA-SIFT (with 10 components).

Mean of geometric errors is smaller than 0.5pixel and max of geometric errors is small in Fig.9. ROBPCA-SIFT20 and ROBPCA-SIFT10 operate well such as SIFT or CLAPCA-SIFT in endoscopic image without the noise.



(a) ROBPCA-SIFT20



(b) ROBPCA-SIFT10

Figure 10: Geometric error in endoscopic image with the noise. (a) using ROBPCA-SIFT (with 20 components). (b) using ROBPCA-SIFT (with 10 components).

Mean of geometric errors is smaller than 0.5pixel and max of geometric errors is not too big in Fig.10 (a). But mean of geometric errors is not smaller than 0.5pixel and max of geometric errors is big in some

cases in Fig.10 (b). ROBPCA-SIFT20 operates well but ROBPCA-SIFT10 does not operate well in endoscopic image with the noise. ROBPCA-SIFT20 is reliable in endoscopic image with the noise.

6 Conclusions

The advantage of the method that was introduced in section.3 is to reduce the dimension. SIFT, CLAPCA-SIFT, and ROBPCA-SIFT are reliable in endoscopic image with the noise. Using 10 principal components out of 128 components may enough in endoscopic image without the noise. However, the case which uses ROBPCA-SIFT is more reliable than the case which uses SIFT or CLAPCA-SIFT in noised endoscopic images. Using 10 principal components out of 128 components may not enough in noised endoscopic image. ROBPCA-SIFT20 is most reliable. The fundamental matrix that encodes the exterior parameters of the camera is computed by using RANSAC and the 3D projective reconstruction is accomplished from only the endoscopic images.

From this research, we claim that there are some products for stereo endoscope. However the 3D information can be extracted reliably by using the prevailing endoscope with the idea of this paper.

7 Acknowledgements

This research was carried out with the support from the Image Analysis Technological Foundation Planning & Development Board of the Korean Institute of Science & Technology Evaluation and Planning, Republic of Korea. (M10429040003-04L2904-00310).

8 References

- [1] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector", *ECCV'02*, vol I, pp 128–142.
- [2] David G. Lowe, "Distinctive image feature from scale-invariant keypoints", *accepted for publication in the International Journal of Computer Vision*, 2004.
- [3] David G. Lowe, "Object recognition from local scale-invariant keypoints", *Proc. of the International Conference on Computer Vision*, Corfu, pp 1150–1157, 1999.
- [4] David G. Lowe, "Local feature view clustering for 3D object recognition", *IEEE Conference on Computer Vision and Pattern Recognition*, Kauai, Hawaii, pp 682–688, 2001.
- [5] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors", *In Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Washington, DC, pp 511–517, 2004.

- [6] S. Frosch Møller, J. von Frese, and R. Bro, "Robust methods for multivariate data analysis", *Journal of Chemometrics*, 19: pp 549–563, 2005.
- [7] M. Hubert, P.J. Rousseeuw, and K. Vanden Branden, "ROBPCA: a new approach to robust principal component analysis", *Technometrics*, 47: pp 64–79, 2005.
- [8] S. Verboven and M. Hubert, "LIBRA: a MATLAB library for robust analysis", *Chemometrics and Intelligent Laboratory Systems*, 75: pp 127–136, 2005.
- [9] Richard Hartley and Andrew Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2003.

Towards nuclear phenotype recognition in single channel fluorescence microscopy images

I. Sintorn¹, L. Bischof¹, R. Lagerstrom¹, M. Buckley¹ and A. Hoffman²

¹CSIRO Mathematical and Information Sciences, Locked Bag 17, North Ryde NSW 1670, Australia.

²Roche Discovery Technologies, Hoffmann-La Roche Inc. 340 Kingsland Street 123/2440 Nutley, NJ 07110

Email: Ida-Maria.Sintorn@csiro.au

Abstract

We describe a robust segmentation technique and a set of features for characterizing nuclear figures in single channel images of DNA stained cell nuclei. The features are specifically designed to differentiate between mitotic figures. They characterize size, shape, symmetry, directionality, and intensity distribution for each figure. We show that they can be flexibly combined to classify nuclei into seven different phenotype classes. The number of classes and how the features are combined in those classes depends on resolution limitations and investigator choices.

Keywords: nuclear phenotypes, mitotic figures, fluorescence microscopy

1 Introduction

High Content Imaging and the advancements made in image analysis research are expanding the techniques available to both quantify and classify cellular phenotypes. This is of great importance in the quest to characterize genes with unknown functions as well as to analyze the effects of small molecules (drug candidates) and to establish cell signalling pathways. Often, this kind of analysis is still performed by visually interpreting (scoring or classifying) large numbers of microscopy images and/or their contents [1, 2, 3, 4]. The importance of better (more specific) and faster image analysis methods for this kind of research is emphasized in for example [5], a recent review about using RNAi to establish the functionality of unknown genes. In [6, 7, 8] examples of automated image analysis solutions to certain kinds of cellular and subcellular compartment phenotyping can be found. There are a number of reasons or factors to why automated image analysis solutions aren't more widely used; complex images (multi-channel, 3D, time series), specific questions where the tools available in general packages aren't enough, poor knowledge of what tools are available and how to use them, user unfriendly software etc.

We have been interested in using automated image analysis to study drug effects on cell cycle regulation and how this relates to the ensuing nuclear phenotypes. This is similar to the work presented in [2] but using only single channel images. The benefit of using only one channel to analyse the

phenotypes is that it leaves two or three channels available on many of the High-content imaging instruments and microscopes for investigators to query their own particular problems. In figure 1, normal as well as some abnormal nuclear phenotypes within the cell-cycle are shown. A mitotic cycle (the sequence of steps in cell division) starts with a normal nucleus (a). The DNA is duplicated and during the metaphase it is concentrated along a line (b). During the telophase and anaphase the two copies of the DNA are pulled towards two opposite poles (c, d), and during cytokinesis the cell splits into two daughter cells and the DNA spreads out in the nuclei to its normal state (e, a). The abnormal nuclear figure (f) shows an apoptotic (programmed cell death) nucleus. A nucleus can in fact turn apoptotic at any stage of the cell cycle. The two abnormal nuclear figures (g, h) show a monastral nucleus and a tripolar mitotic nucleus, respectively.

The goal of the research presented here is to establish a core set of image measures that can be used to accurately classify a nucleus into seven distinct phenotypes for a given experimental procedure. The procedure should also be easily applied, with minimal modification of parameters, to similar image sets generated using a different high-throughput imaging platform, cell type, objective and/or staining.

There are several approaches to tackling this problem, all with different advantages and disadvantages. One popular method is to extract a large

number (often in the order of 100's) of generic texture and shape measurements from an extensive training set of images containing the phenotypes of interest, see for example [6, 7]. A classifier can then be derived through any of the standard data mining approaches (such as neural networks, linear or non-linear statistical classifiers) to classify these measurements into the training class phenotypes. This approach has the advantage that no custom image analysis development is required. However, the use of a large number of measures has two disadvantages: it is difficult for the biologist to understand the significance of the individual measures; and a large training set must be used to avoid over-fitting the classifier to the training set and thus limiting its performance on new images.

Our approach is to custom design an image analysis algorithm to extract a small set of measures to recognise specific nuclear phenotypes. The algorithm design is based on two factors: prior knowledge of the nuclear shapes to be recognised, derived from an understanding of the biology; and a small set of training images to validate the measures. There are several benefits of this approach. One is that it is easier for a biologist to understand the significance of each of the measures. As a consequence, it is possible to make simple combinations of these measures to create classification rules for each nuclear shape without the need for extensive data mining. It is even possible to generate measures to detect phenotypes for which there are no training samples. For instance, a feature designed to measure the number of poles of a metaphase nucleus can detect the normal 2-pole figure and abnormal 3-pole figure, see figure 1 (g, h), for which we have training samples, but it can also be expected to detect 4-pole nuclear figures for which we have no examples.

In Section 2 we explain the method used to segment all cell nuclei. We then present the binary and intensity based features designed to differentiate between different nuclear phenotypes. In Section 4 we combine the features to separate between seven different nuclear phenotypes.

2 Pre-processing and Segmentation

A small (3×3) gaussian filter was used to smooth the images which were then background corrected using a top hat filtering with a structuring element larger than the largest expected nuclear width. The image was thereafter thresholded using a method that finds a global (image-wide) threshold by utilising the gradient strength information, similar to [9]. A bivariate histogram of the gradients (approximated by applying a set of Sobel operators) and the grey-levels is calculated.

The average gradient strength for each grey-level is interpreted as a histogram. The grey-level that corresponds to an input quantile parameter serves as the threshold.

Touching nuclei were next separated using an internal distance transform (DT) [10] and a watershed transform [11], a very common approach in cell nuclei segmentation.

The global threshold segmentation gives good boundaries for the majority of the nuclei in the image but not for the particularly bright ones, such as the mitotic phenotypes (which are brighter due to condensed DNA). For these objects (with high internal average intensity and/or intensity variation), a local (per-nucleus) threshold at their median grey-value was used to ensure that the shape of mitotic figure was accurately detected. Objects that were too dark (out of focus) or too small (noise or cell debris) to be of interest were discarded at this stage.

3 Feature Extraction

A number of standard features were combined and some more specific ones were developed to characterise some common nuclear phenotypes important for monitoring cell cycle regulation. Nuclei of normal and non-dividing cells vary quite a bit in shape and size but are in general somewhat ellipsoidal, figure 1 (a). Apoptotic cells on the other hand are characterized by smaller size, very high intensity and circular shape, figure 1 (f). Metaphase nuclei are bright and elongated in a rectangular rather than ellipsoidal fashion, figure 1 (b). Normal anaphase nuclei are also bright and elongated like the metaphase nuclei but smaller and also paired, figure 1 (c, d). Monastral nuclei are a kind of abnormal anaphase nuclei which are bright and ringshaped, figure 1 (g), and tripolar nuclei (Y-shaped), are another type of abnormal anaphase nuclei, figure 1 (h). There can also be other abnormal multipolar nuclei with 4 (X-shaped) or more poles. We use a combination of features for characterising the objects shape, intensity distribution, orientation and symmetry.

As mentioned in section 2, nuclei are segmented by first a global threshold followed by an optional local threshold for especially bright objects. A series of features are extracted for all segmented objects. A list of the features and a description of the type of information they summarise are found in table 1. Several of them, **size**, **intensity**, **intensity variability**, **elongation**, **rectangularity**, **circularity-1**, **circularity-2** are very commonly used intensity and shape features. Only the few that we custom designed need further explana-

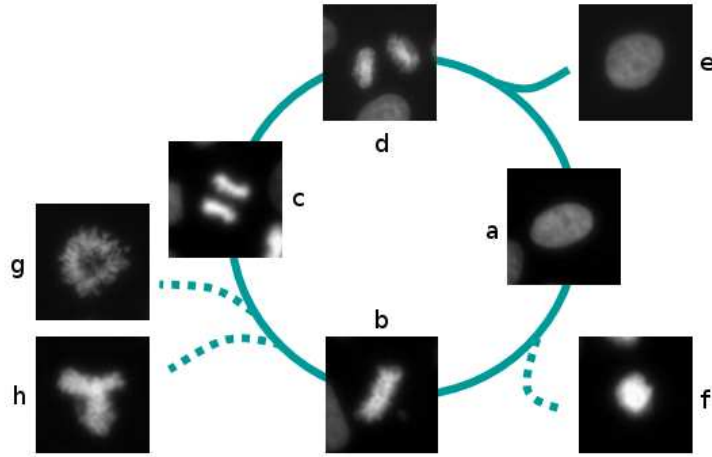


Figure 1: An illustration of the cell cycle with images showing some different normal (solid lines) and abnormal (dashed lines) nuclear figures. See text for further details.

tion. **Intensity distribution** is designed to detect the radial intensity distribution of monastral figures by measuring the average intensity difference between the outer ring and the inner disc of a region (divided into 2 regions by a central circle of diameter $1/3$ of the diameter of the whole region). **Convexity** measures the average distance of the object boundary from its convex hull. **Pair-wise angle** quantifies the orientation of two neighbouring objects. It measures the angular differences between a line connecting the two object centres and the minor axis of each object, with the greater difference being used as the pair-wise angle measure. This is used to distinguish parallel and non-skewed object pairs, see figure 2 (bottom row). The two rotational symmetry measures, for **3-fold and 4-fold symmetry** respectively, are based on how much the intensity distribution is concentrated (as measured by the angular variance) for each pixel (transformed to polar coordinates) when multiplied by 3 or 4, respectively. This is similar to the symmetry descriptors in [12], but using intensity rather than gradient information. See figure 2 (top row) for examples of figures these features can distinguish (note that the 4-polar figure is artificially constructed from two metaphase figures for illustrative purposes).

4 Phenotype Classification Example

To test the performance of our features, we used a set of 26 fluorescence images of DNA stained (using Hoechst 33258) HeLa cells that had been exposed to a number of known and exploratory cell cycle affecting agents. The images were acquired on a Zeiss Axiovert fluorescence microscope

Table 1: Features

Feature	Description
Size	Area
Intensity	Mean intensity
Intensity variability	Intensity std
Intensity distribution	Peripheral-central intensity
Elongation	Major/minor
Rectangularity	Major*minor/Area
Circularity-1	$\frac{Perimeter^2}{Area * 4 * \pi}$
Circularity-2	$\frac{(Major + minor)^2 * \pi}{16 * Area}$
Convexity	Mean concavity depth
Pair-wise angle	Minor axes vs centreline angle
3-fold symmetry	3-fold angular sum
4-fold symmetry	4-fold angular sum

at 20X magnification, and a typical example is shown in figure 3. The object features were combined to classify objects into seven different nuclear cell cycle phenotypes; *normal*, *metaphase*, *ana-telophase*, *monastral-like spindles*, *multipolar (3- and 4-polar)*, *apoptotic*, and *remaining*, see figure 1. In this application example, the *remaining* class simply consists of nuclear phenotypes that don't fit in any of the other six classes.

Since (so far) only a set consisting of 26 test images was available to test our features, all parameters and feature thresholds used for classifying the objects were set manually by examining the range of feature responses for a few nuclei of each class. A tree-like classifier was then constructed to classify the nuclear figures into one of the seven classes. The heuristic decision rules derived were as fol-

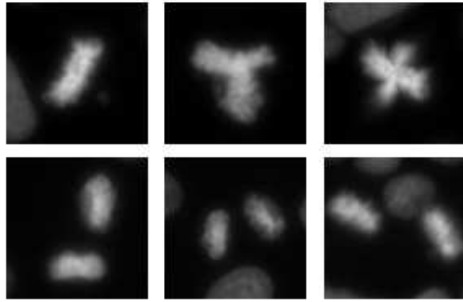


Figure 2: Nuclear figures that can be differentiated using 3- and 4- fold symmetry (top row), and pairwise angle (bottom row).

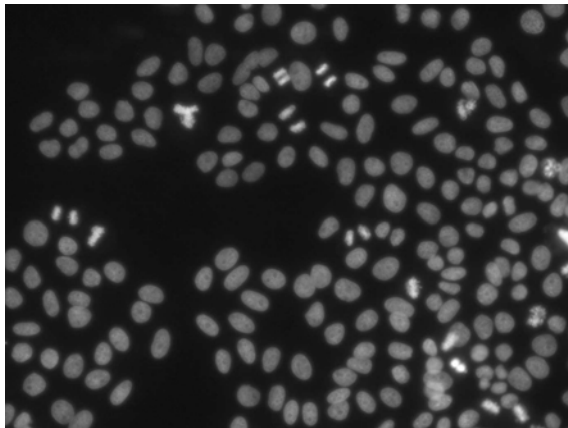


Figure 3: An example of a typical input image.

lows. First, too large objects (with **size** greater than a specified maximum area) were assigned to the *remainder* class. Next *monastral* figures were identified if an objects **Size** and **Intensity distribution** features were within a specified range. The nuclei with normal intensity were classified as *normal* if they were fairly circular (**circularity-1**) or ellipsoidal (**Elongation**), and the ones that were not within range were put in the *remainder* class. For the bright nuclei, the *multi-polar* figures (very few in this set) were first found by the two **symmetry** measures. Bright figures were classified as *ana-telophase* if they were small (**size**), elongated (**elongation**), and paired in an appropriate configuration (**pair-wise angle**), and as *metaphase* figures if they were fairly small, elongated, rectangular (**rectangularity**) and convex (**convexity**). *Apoptotic* figures were found based on high intensity and circular shape (**circularity-1 and -2**), and finally the bright figures that were left were set to the *remainder* class. A gallery of figures classified to the seven classes are shown in figure 4. Each row represents a specific class and the stars mark figures that are misclassified.

In total there were 1899 nuclear figures in the images, of which our segmentation method

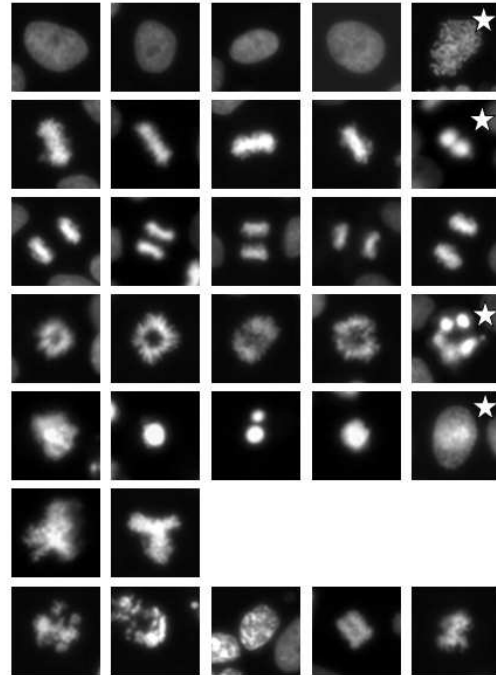


Figure 4: Examples of figures classified as *normal* (row 1), *metaphase* (row 2), *ana-telophase* (row 3), *monastral like spindles* (row 4), *apoptotic* (row 5), *multi-polar* (row 6), and *remainder* (row 7). The figures marked with stars are examples of false positives.

correctly segmented 1868 (99%). The number of correctly classified/false positives/false negatives figures of classes were: *normal* (1736/3/5); *metaphase* (46/5/0); *ana-telophase* (17/0/0); *monastral-like spindles* (6/3/0), *multipolar* (2/0/0); *apoptotic* (77/5/3).

5 Conclusion & future work

We have developed a two-step segmentation procedure that very accurately segments nuclei objects even though they have widely varying brightness. The small set of object measures that we designed specifically to discriminate nuclear phenotypes, are intuitively meaningful to a biologist and can be readily combined into an *ad hoc* set of rules to classify each object. This means that the biologist can modify these rules for each new experiment without having to perform complex data analysis. The classification accuracies achieved are good, although there were very few objects of certain phenotypes to fully validate our approach.

It should be mentioned that the current image resolution (captured using a 20X objective) is close to the minimum for which this set of features will work reliably. Some shape features are rather unstable when the nuclear phenotypes are very small

(one pixel on the width or height can make a noticeable difference in these features).

Future work will involve the acquisition of more data (larger sets of the same set-up as well as other stains, microscopes etc). This will enable us to refine the classifier (get a better understanding of within-phenotype variation), to develop a method to derive the parameters automatically, and to generate new features for other phenotypes to get a more general nuclear phenotype classifier.

References

- [1] N. J. Quintyne, J. E. Reing, D. R. Hoffelder, S. M. Gollin, and W. S. Saunders, "Spindle multipolarity is prevented by centrosomal clustering," *Science*, vol. 307, 2005.
- [2] M. Bettencourt-Dias, R. Sinka, A. Mazumdar, W. G. Lock, F. Balloux, P. J. Zafirooulos, S. Yamaguchi, S. Winter, R. W. Carthew, D. Jones, L. Frenz, and D. M. Glover, "Genome-wide survey of proteins kinases required for cell cycle progression," *Nature*, vol. 432, no. 23/30, pp. 980–987, 2004.
- [3] L. Pelkmans, F. Eugenio, H. Grabner, M. Hannus, B. Habermann, E. Krausz, and M. Zerial, "Genome-wide analysis of human kinases in clathrin- and caveolae/raft-mediated endocytosis," *Nature*, vol. 436, pp. 78–86, 2005.
- [4] P. Holmberg, S. Stenmark, and M. Gullberg, "Differential functional interplay of *togp/xmap215* and the kini kinesin *mcak* during interphase and mitosis," *The EMBO Journal*, vol. 23, no. 3, pp. 627–637, 2004.
- [5] F. Fuchs and M. Boutros, "Cellular phenotyping by *rnai*," *Briefings in functional genomics and proteomics*, vol. 5, no. 1, pp. 52–56, 2006.
- [6] Z. E. Perlman, M. D. Slack, Y. Feng, T. J. Mitchison, L. F. Wu, and S. J. Altschuler, "Multidimensional drug profiling by automated microscopy," *Science*, vol. 306, pp. 1194–1198, 2004.
- [7] T. R. Jones, A. E. Carpenter, P. Golland, and D. M. Sabatini, "Methods for high-content, high-throughput image-based cell screening," in *MIAAB (Microscopic Image Analysis with Applications in Biology) 2006 Workshop Proceedings*. preprint on web (<http://jura.wi.mit.edu/cellprofiler/papers.htm>).
- [8] J. N. Harada, K. E. Bower, A. P. Orth, S. Callaway, C. G. Nelson, C. Laris, J. B. Hogenesch, P. K. Vogt, and S. K. Chanda, "Identification of novel mammalian growth regulatory factors by genome-scale quantitative image analysis," *Genome Research*, vol. 15, pp. 1136–1144, 2005.
- [9] J. S. Weszka and A. Rosenfeld, "Histogram modification for threshold selection," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, pp. 38–52, 1979.
- [10] G. Borgefors, "Distance transforms in arbitrary dimensions," *Computer Vision, Graphics, and Image Processing*, vol. 27, pp. 321–345, 1984.
- [11] S. Beucher, "The watershed transformation applied to image segmentation," *Scanning Microscopy*, vol. 6, pp. 299–314, 1992.
- [12] C. Sun, "Fast recovery of rotational symmetry parameters using gradient orientation," *Optical Engineering*, vol. 36, no. 4, pp. 1073–1077, 1997.

The Waikato Range Imager

M. J. Cree¹, A. A. Dorrington¹, R. M. Conroy¹, A. D. Payne¹, and D. A. Carnegie²

¹Department of Engineering, University of Waikato, Hamilton.

²School Physical & Chemical Sciences, Victoria University of Wellington, Wellington.

Email: cree@waikato.ac.nz

Abstract

We are developing a high precision simultaneous full-field acquisition range imager. This device measures range with sub millimetre precision in range simultaneously over a full-field view of the scene. Laser diodes are used to illuminate the scene with amplitude modulation with a frequency of 10 MHz up to 100 MHz. The received light is interrupted by a high speed shutter operating in a heterodyne configuration thus producing a low-frequency signal which is sampled with a digital camera. By detecting the phase of the signal at each pixel the range to the scene is determined. We show 3D reconstructions of some viewed objects to demonstrate the capabilities of the ranger.

Keywords: Range imaging, imaging lidar, heterodyne, image intensified

1 Introduction

The Waikato Range Imager is a full-field imaging lidar system that is capable of producing high resolution images by simultaneously measuring the range in the field of view as seen by each pixel. The ranger is capable of acquiring sub-millimetre precision in range under optimal conditions for a full-field in 10 seconds. In this paper we give an overview of the system and present some recent range images and applications that we have investigated.

2 Imaging Lidar

Image ranging systems can usefully be classified as laser point scanning or full-field (simultaneous) image acquisition. The high precision ranging and x - y positioning of the laser scanner is obtained by moving a laser dot over a field of interest, however the acquisition times can be very long. Such systems are numerous in the literature and in models commercially available [1]. Full-field acquisition, or imaging lidar as it is sometimes called, offers the potential of fast and precise measurement over the whole field of view, but remains somewhat in its infancy with few systems demonstrated with varying degrees of success [2, 3, 4, 5, 6, 7, 8]. Despite the variety of implementation methods, there is much commonality in operating principles and hardware configurations. The operating principle is the expansion of time-of-flight point laser rangers to operate simultaneously over a full field of view. The Waikato Range Imager falls into the imaging lidar category.

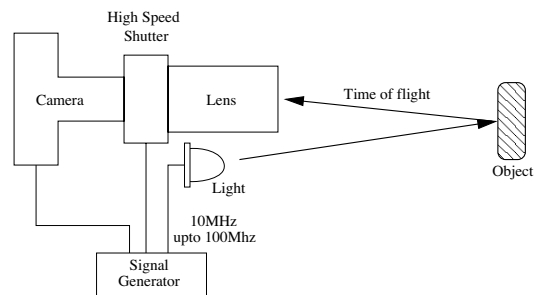


Figure 1: Basic components of an imaging lidar system.

The basic means of operation of imaging lidar is as follows (also see figure 1): A modulated light source illuminates the scene and the light is scattered by objects in the scene to be collected by a camera system. The camera system incorporates a high speed (non-mechanical) shuttering system to modulate the intensity of received light. The major difference in imaging lidar systems lies in the high speed shuttering mechanism and in the modulation control signals. The most common shuttering mechanism is an image intensifier with high speed photocathode modulation capability. Image intensifiers have a number of disadvantages thus there is a move to integrate the shuttering mechanism into custom image sensors. A number of these types of sensors have been described, but they are currently limited by low spatial resolution [4, 8].

The modulation control philosophies can be grouped as pulsed, homodyne or heterodyne. In pulsed systems the illumination source and the high speed shutter are controlled with a single pulse in the nano-seconds region [6]. The scattered

light from the scene entering the camera is time delayed due to the path length travelled. The received pulse from a close object will align well with the shutter pulse and an intense signal is received. A received pulse from a far away object will not coincide well with the shutter pulse and a weak signal is received. The brightness of a pixel is therefore correlated with range. Homodyne systems are similar, but a continuous modulation in the 10 MHz to 100 MHz region of the illumination and shutter is used to improve SNR and reduce the requirements of high-speed electronics. Some decoding of the signal is required to derive actual range values from intensity, and such schemes as quadrature or phase-shift keying are often used. Nevertheless systems based on the pulsed or homodyne philosophy have range precision that is limited by the dynamic range of the camera (often a CCD) and these systems typically achieve at best centimetre precision over a distance of less than five metres.

Heterodyne systems are different from homodyne systems in that the modulation frequencies applied to the illumination source and the high speed shutter differ very slightly. The mixing process at the shutter produces a low frequency beat at the difference of the frequencies of the illumination source and the high speed shutter. The phase delay of the received light (hence the range information) is preserved on the heterodyne beat signal. Thus a scene observed by the camera appears to flash, with close objects flashing at a different time to those far away. Range information can be obtained by calculating the beat signal phase (over time) for each pixel. The range precision is therefore limited by the accuracy with which the phase of the beat signal can be measured. This is the approach used in the Waikato Ranger Imager.

3 The Waikato Range Imager

Figure 2 shows the Waikato Range Imager. The illumination source is a bank of four laser diodes (658 nm) rated at 80 mW for continuous output. These are fibre optically coupled to illuminate the scene from a ring surrounding the camera lens. This scheme helps to ensure that the path length from the light source to scene can be calculated as originating from the optic axis about the plane of the lens focal point. The light from the ends of the fibre optics is allowed to disperse to illuminate the whole scene.

Like many other imaging lidar arrangements [2, 6, 7] the Waikato Range Imager employs an image intensifier as the high speed shutter. A Photek 25 mm single microchannel plate (MCP) image intensifier is used. The 25 mm diameter on the

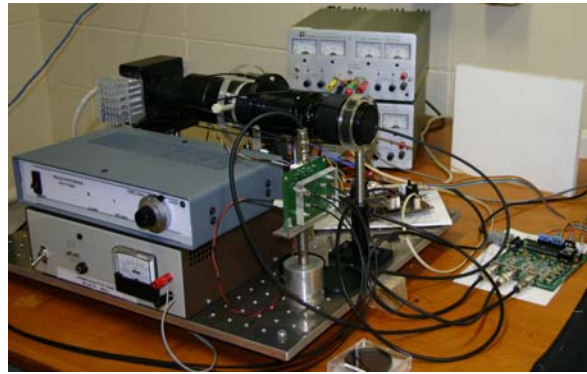


Figure 2: The University of Waikato Range Imager

entrance window allows easy imaging of the scene with standard F-mount lenses. Image intensifiers are often used for high speed photography by switching the MCP voltage on and off in a very short single pulse. This provides extremely good contrast between the on and off shutter states. Because the MCP voltage is approximately 700 V and we require continuous switching at frequencies up to 100 MHz, switching the MCP voltage is not feasible. We therefore choose to switch the photocathode voltage with a 50 V amplitude signal. Image quality is affected by the photocathode voltage and as the voltage passes through the low voltage régime some blurring of the image formed by the image intensifier occurs. We use square wave modulation to minimise the transit time through low voltages.

The illumination wavelength is not optimal for the image intensifier as the S20 photocathode sensitivity at 658 nm is only 40% compared to its 450 nm peak. The laser diodes were purchased for their combination of low cost and high power. There are other issues with using image intensifier technology and they are discussed in a companion paper [9].

A Dalsa Pantera TF 1M60 digital video camera is used to acquire video sequences of the beat signal appearing on the phosphor screen of the image intensifier. This camera is a 1 megapixel, 60 fps, true 12-bit camera with good sensitivity. Since the image intensifier is the resolution limiting component we run the camera in the 2×2 binning mode providing 512×512 pixels at up to 100 fps. This provides two advantages: better photon statistics (hence better SNR) and a higher video sampling rate (hence quicker estimation of phase). The camera is coupled to the image intensifier with a high quality fixed focal length relay lens that has a working distance of 15 mm. Better collection of light of an order of magnitude would result from using direct fibre optic coupling between the image intensifier and camera CCD, however at this stage we prefer the flexibility of using the relay lens.

The modulation signals for the image intensifier and illumination source and the frame trigger for the camera are generated by three direct digital synthesis (DDS) chips driven by a common digital clock source. Operating multiple synthesisers from the same digital clock source produces highly accurate relative frequencies as any drift is common to all outputs. To enable absolute range measurements the phase of the beat signal generated by mixing of the received light of an object at zero range in the image intensifier is required as a reference. Measuring the phase difference between the two outputs is a challenging task due to the high frequencies involved and the resolution required. The distance to phase relationship is given by

$$\theta = \frac{4\pi fd}{c} \quad (1)$$

where θ is the phase, f is the modulation frequency, d is the distance being measured and c is the speed of light. Hence to obtain millimetre range precision using 100 MHz modulation requires the phase to be measured to a precision of less than 4 mrad. The reference phase precision must be better than this and be known to 12 bit resolution (i.e. 1.5 mrad). This is achieved by using the third DDS to produce a synchronised signal at the low frequency difference of the first two outputs; its phase can be directly measured to provide the reference phase difference. This signal is also used to provide the camera frame trigger keeping it synchronised with the rest of the system allowing the scene to be sampled at an exact multiple of the low frequency beat signal.

The availability of the beat reference and frame grabber signals has also allowed the ability to switch off the image intensifier to blank the view during CCD readout. This is important as the CCD continues to integrate the received light even during readout thus smearing scene data down columns of the CCD [10]. This can lead to contamination of the phase of the beat signal along columns of the CCD. By switching off the image intensifier during CCD readout this problem is completely eliminated.

4 Signal Processing

Video sequences are collected of a scene over time with the ranger system. Each pixel is analysed in time for the beat signal. In earlier incarnations of the ranger, in which the frame grabbing was not synchronised to the beat signal, Fourier analysis was used to estimate the phase of the beat signal [11, 5]. Now that the hardware has been improved so that the frame grabber is precisely synchronised to the beat frequency an inner product of a sine wave of the known beat frequency

with the signal at a pixel suffices to isolate the signal to calculate the phase and, hence, the range. This approach affords a significant advantage: the signal processing can be implemented in real time, thus eliminating the need to save video data, other than maintaining a buffer for the current frame.

There is a potential problem for the naïve: the image intensifier has a non-linear response thus there are harmonics on the signal. It is not possible to low-pass filter the signal (the beat on the image intensifier) before sampling (frame-grabbing) therefore any harmonics above the nyquist limit are aliased. If an aliased harmonic happens to land at the fundamental frequency after sampling then it can contaminate the phase estimation thereby reducing range precision. Not only must there be an integer number of beat cycles in the sample period for inner-product processing, it is also important to choose the sampling frequency and the beat frequency to have no common factors [12]. This is contrary advice to that normally given in phase measurement problems such as occurs in interferometric phase shifting for profile measurement. There the signal is cleanly sinusoidal.

5 Results

A number of objects were imaged to show the capability of the Waikato Range Imager. A steel block of height 100 mm and width 70 mm was imaged as a first example (see figure 3). A second example is a wheel of diameter 175 mm (see figure 4). Ideal imaging conditions were used; in particular the blocks were painted matte white to improve signal detection and reduce specular reflections. A modulation frequency of 78 MHz, beat frequency of 1 Hz and camera frame-grabbing frequency of 29 Hz was used to capture these two examples.

For both examples we show both a photograph of the object and the three-dimensional rendering reconstructed from a single view of range data of the object. Details such as the 2 mm high ridges on each spoke of the wheel and the sharp edges of the block are clearly visible. Unfortunately the chosen visualisation method tends to accentuate the measurement noise. An in-depth discussion of the error sources is beyond the scope of this paper, but can be found in references [9, 13, 14]. In previous experiments under the same operating conditions we have demonstrated 0.4 mm precision for ranging at the one standard deviation uncertainty level over distances of up to 6 m [13]. To estimate the precision achieved in the range images we have fitted a plane with a least-square fitting approach to a number of small areas of various faces of the block (fig. 3) and used the mean of the



Figure 3: Photograph (top) and 3D reconstruction from range data (bottom) of a block.

Table 1: Precision achieved for various small areas of the block

Size of Area (pixels)	Precision (mm)
16×16	0.385
18×18	0.332
16×16	0.345
20×20	0.342

residuals as an estimate of precision. The results are listed in table 1 and indicate that the 0.4 mm precision is being achieved in these examples.

As a third example we show the range image (figure 6) and the three-dimensional reconstruction (figure 7) of ‘Stumpy’ – a garden gnome (see figure 5). Stumpy was imaged with 80 MHz modulation frequency, 1 Hz beat frequency and 29 fps sampling rate for a period of 10 s. For this case Stumpy was imaged ‘as is’ and one can see the noisier reconstruction (see figure 7) resulting from



Figure 4: Photograph (top) and 3D reconstruction from range data (bottom) of a wheel.



Figure 5: Stumpy: the garden gnome under investigation (for various unresolved crimes).



Figure 6: Range image of Stumpy. Increasing intensity represents increasing range to the scene.

the dark areas (such as the spade) where a poor signal is received. Despite the poor signal (signal amplitude less than 2% of that of the bright regions) the general shape of the spade is nevertheless reconstructed. Note the detail of features detected where good signal is received, such as the eye lashes that are less than 2mm deep and the ridges in the ears that are less than 1mm deep. The Waikato Range Imager measures intensity at the same time as measuring range and in figure 8 we show a visualisation in which the intensity data is overlaid the 3D reconstruction.

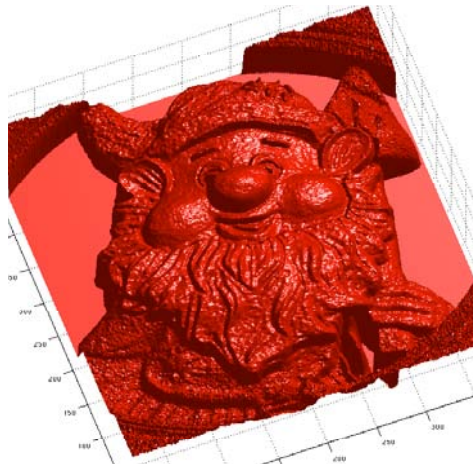


Figure 7: 3D reconstruction of Stumpy.

6 Discussion

We have demonstrated precision, operational distance and spatial resolution all at the upper end of the scale compared to other solid-state range imagers. Furthermore, we have demonstrated all those characteristics simultaneously, which, to our knowledge, no other group has demonstrated.

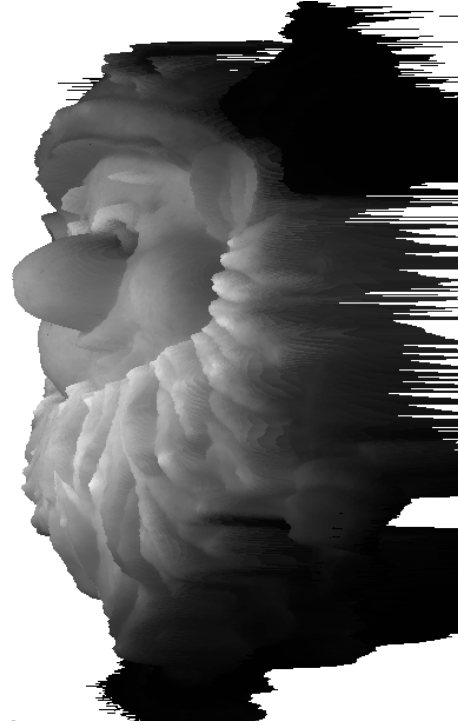


Figure 8: 3D reconstruction of Stumpy overlaid with intensity information.

Even though we have achieved one of our main aims with the Waikato Range Imager, namely high precision simultaneous full-field ranging, there remain a number of factors that can be improved. The high precision has been obtained at the expense of time; acquisitions reported here take 10s. It would be nice to get the acquisition time below 1s for near real time imaging. Currently one can only reduce the acquisition time at the expense of range precision, for example, we have demonstrated approximately 1cm precision with 1s acquisitions.

Our reported precision is only for repeated range measurements from a single pixel. There remain systematic errors across the field of view. These are due to geometrical distortion for off axis viewing, lens distortions and image intensifier distortions. In principle they can be calibrated for. Geometrical distortion corrections and lens calibrations are well reported and could be easily applied. The behaviour of the image intensifier is less well described and we report some of our own investigations in a companion paper [9]. One important problem is iris in the image intensifier. This occurs because the switching off (or on) of the image intensifier proceeds from a ring at its outside and progresses over time towards the centre of the image intensifier. This leads to a phase delay in the measurements at the centre of the field of view compared to those at the periphery. Thus range reconstructions report inflated range values as a

function of radial distance from the optical axis in the field of view.

The image intensifier requires high voltages and, with its power supplies, is bulky. It remains the greatest obstacle to miniaturising the technology. A few custom image sensors that incorporate the high speed shuttering function in the sensor have been described but they are currently of low resolution and typically only achieve 1 cm range precision [8, 4].

7 Acknowledgements

AAD is funded by a FRST Postdoctoral Fellowship. ADP and RMC both acknowledge the receipt of a TEC Bright Futures PhD Scholarship. The authors are grateful to WaikatoLink Ltd. for funding of hardware and studentships. The Waikato Imager Ranger is protected by international and New Zealand patents.

References

- [1] F. Blais, "Review of 20 years of range sensor development," *J. Elect. Im.*, vol. 13, pp. 231–240, 2004.
- [2] S. Christie, S. L. Hill, B. Bury, J. O. Gray, and K. M. Booth, "Design and development of a multi-detecting two-dimensional ranging sensor," *Meas. Sci. Tech.*, vol. 6, pp. 1301–1308, 1995.
- [3] B. L. Stann, M. M. Giza, D. Robinson, W. C. Ruff, S. D. Sarama, D. R. Simon, and Z. G. Sztankay, "Scannerless imaging lidar using a laser diode illuminator and FM/cw radar principles," *Proc. SPIE*, vol. 3707, pp. 421–431, 1999.
- [4] P. Gulden, M. Vossiek, P. Heide, and R. Schwarte, "Novel opportunities for optical level gauging and 3-D-imaging with the photo-electronic mixing device," *IEEE Trans. Instr. Meas.*, vol. 51, pp. 679–684, 2002.
- [5] D. A. Carnegie, M. J. Cree, and A. A. Dorrington, "A high resolution full-field range imaging system," *Rev. Sci. Instr.*, vol. 76, p. 083704, 2005.
- [6] J. Busck and H. Heiselberg, "Gated viewing and high-accuracy three-dimensional laser radar," *Appl. Opt.*, vol. 43, 2004.
- [7] M. Kawakita, K. Iizuka, H. Naruhito, I. Muzuno, T. Kurita, T. Aida, Y. Yamanouchi, H. Mitsumine, T. Fukaya, K. Kikuchi, and F. Sato, "High-definition real-time depth-mapping TV camera: HDTV axi-vision camera," *Opt. Express*, vol. 12, pp. 2781–2794, 2004.
- [8] S. B. Gokturk, H. Yalcin, and C. Bamji, "A time-of-flight depth sensor—system description, issues and solutions," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'04)*, vol. 3, pp. 35–43, 2004.
- [9] A. D. Payne, A. A. Dorrington, M. J. Cree, and D. A. Carnegie, "Image intensifier characterisation," in *Image and Vision Computing New Zealand (IVCNZ'06)*, (Gt. Barrier Island, New Zealand), November 2006. Accepted.
- [10] A. A. Dorrington, M. J. Cree, and D. A. Carnegie, "The importance of CCD readout smear in heterodyning imaging phase detection applications," in *Image and Vision Computing New Zealand (IVCNZ'05)*, (Dunedin, New Zealand), pp. 73–78, 2005.
- [11] M. J. Cree, A. A. Dorrington, and D. A. Carnegie, "A heterodyning range imager," in *IAPR Conference on Machine Vision Applications*, (Tsukuba Science City, Japan), pp. 80–83, 2005.
- [12] A. A. Dorrington, D. A. Carnegie, M. J. Cree, and A. D. Payne, "Selecting signal frequencies for best performance of Fourier-based phase detection," in *12th Electronics New Zealand Conference (ENZCON'05)*, (Auckland, New Zealand), pp. 189–193, 2005.
- [13] A. A. Dorrington, M. J. Cree, A. D. Payne, R. M. Conroy, and D. A. Carnegie, "Achieving sub-millimetre precision with a solid-state full-field heterodyning range imaging camera," *Meas. Sci. Tech.* Submitted.
- [14] A. A. Dorrington, M. J. Cree, D. A. Carnegie, A. D. Payne, and R. M. Conroy, "Heterodyne range imaging as an alternative to photogrammetry," in *SPIE 6491 – Videometrics IX*, (San Jose, CA), February 2007. Abstract accepted.

Digital Speckle Photogrammetry

Yizhe Lin¹, John Morris¹, Quentin Govignon² and Simon Bickerton²

¹Department of Computer Science,

²Centre for Advanced Composite Materials,

The University of Auckland, Private Bag 92019, Auckland 1000, New Zealand

Email: j.morris@auckland.ac.nz

Abstract

Monitoring of the resin infusion process used to form advanced composite materials requires estimation of displacements of $\sim 0.05\text{mm}$ over an area of $\sim 0.1\text{m}^2$. Optical artefacts (mainly specular highlights) prevented the use of sub-pixel estimations in traditional stereophotogrammetry. However, we were able to adapt digital speckle photogrammetry to obtain the required displacement accuracy. Our modified technique uses two verging axis cameras but measures displacements from image to image of the *same* camera by measuring the phase shift in the Fourier transform from one image to the next. A sparse set of reliable correspondences between the left and right images permitted triangulation to obtain absolute depths - as in conventional stereophotogrammetry. We achieved a depth resolution of $\sim 0.01\text{mm}$ at a distance of $\sim 1\text{m}$ and over an area of $\sim 0.1\text{m}^2$ using two 8Mpixel cameras.

Keywords: Digital speckle photogrammetry, stereo vision, resin infusion

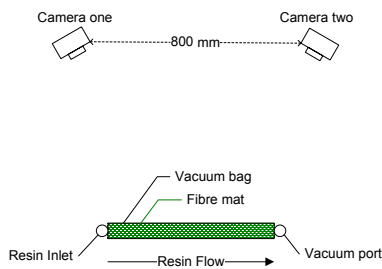


Figure 1: Experiment layout

1 Introduction

This work was inspired by a desire to fully understand the dynamics of resin infusion. The resin infusion process is a low cost method for manufacturing large composite parts with aircraft quality: composite materials are formed by placing a fibre mat inside a plastic bag and draining resin through the bag by applying a vacuum to one end. However, this presents some challenges in reliability and repeatability. A key aim was to be able to measure the bag deformation and compare it with a numerical model.

Contact free non-intrusive measurement and an accuracy of about 0.05mm over an area of 0.1m^2 is needed. Previously, deformations were measured using linear vertical displacement transducers (LVDT) and laser gauges, allowing measurement at a small number of points. Moreover, LVDT requires contact and laser measurements are perturbed by bag surface wrinkles since laser light focuses at a single point.

A preliminary experiment used a verging axis stereo system, see Figure 1. A wide baseline and high reso-



Figure 2: Experimental Rig

lution cameras (Canon 20D, 8Mpixel sensors, $f=50\text{mm}$ lens, see Figure 2) allowed for a 0.3mm accuracy at a distance $\sim 1\text{m}$ without sub-pixel estimation. We could have adopted a sub-pixel estimation scheme, but unstable matching between stereo pairs made this unreliable. The vacuum bag is slightly reflective plastic. This led to different appearances of corresponding regions. Applying a coat of paint alleviated specular effects, but the difference was still significant (see Figure 3). The large view angle also led to significant perspective distortion. Although first order radial lens distortion was largely corrected, residual errors were also still too large to satisfy accuracy requirements.

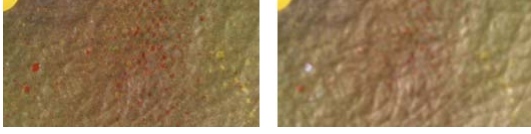


Figure 3: Effect of specular reflections

To overcome the correspondence problem, a modified stereo speckle photography technique was implemented. It traced the relative pattern displacements in the left image sequence and right image sequence, rather than making a left to right correlation. Since all images in the left or right sequence are taken by the same camera, for a small patch, the specular effect, perspective distortion, and lens distortion are almost constant. Therefore, the displacements can be closely approximated as pure translations of patch patterns.

2 Stereo digital speckle photography

Speckle photography is widely used in solid and fluid mechanics [1]. The basic idea is to compare a pattern on the object surface before and after the deformation. Chen *et al.* developed a digital speckle displacement system that integrated optical speckle photography into a process which used correlation in the frequency domain [2]. Synnergren and Sjö Dahl described a stereoscopic digital photography system that permitted 3D displacement measurements [3].

Stereo has advantages over single camera speckle photography. With a single camera, the out-of-plane deformation component is lost and the measured in-plane components are themselves contaminated by perspective errors [4].

Our method is based on that of Sjö Dahl [3]. However, the canonical parallel axis stereo configuration was replaced by a high resolution convergent camera system for higher accuracy. Also, the rather lengthy calculation was simplified by a triangulation procedure.

Figure 4 shows our system. A calibrated convergent axis camera system was used to monitor the vacuum bag during infusion. Before deformation, a pair of reference images were taken by each camera and a set of correspondences were established. During deformation, two sequences of images were taken. The relative displacements to the reference images were calculated on a left-left and right-right comparison basis. Having collected all the initial states and the displacement vectors, the 3D deformations were calculated by triangulating corresponding pairs formed by their initial positions plus displacement vectors.

2.1 Displacement measurement

Given a unique pattern of the object surface, its corresponding region in a subsequent image can be found by

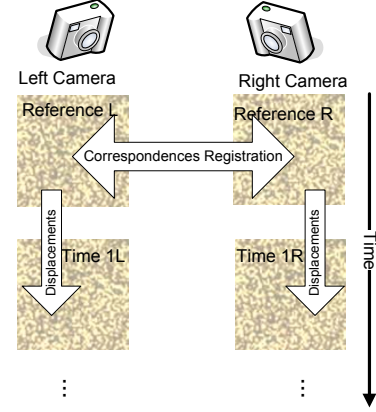


Figure 4: Speckle photography system

cross correlation of the original pattern with the subsequent image.

Consider matching a pattern, $w(x, y)$, of size $J \times K$ in an image, $f(x, y)$, of size $M \times N$, the correlation between $w(x, y)$ and $f(x, y)$ is[5]:

$$c(s, t) = \sum_x \sum_y f(x, y)w(x - s, y - t) \quad (1)$$

where $s = 0, 1, 2, \dots, M - 1$, $t = 0, 1, 2, \dots, N - 1$. The position of the maximal value in $c(s, t)$ indicates where $w(x, y)$ matches $f(x, y)$.

Fortunately, correlation can be computed in the frequency domain if f and w have the same size. Denote the correlation between $f(x, y)$ and $w(x, y)$ as $f(x, y) \circ w(x, y)$, then[5]:

$$f(x, y) \circ w(x, y) \Leftrightarrow F^*(u, v)W(u, v) \quad (2)$$

where $F(u, v)$ and $W(u, v)$ are the Fourier Transforms (FT) of $f(x, y)$ and $w(x, y)$.

Eq 2 shows that correlation in the spatial domain can be obtained by taking the inverse transform of $F^*(u, v)W(u, v)$ in the frequency domain:

$$c(s, t) = \mathbf{F}^{-1}\{F^*(u, v)W(u, v)\} \quad (3)$$

where \mathbf{F}^{-1} is the inverse FT operator.

The Discrete Fourier Transform (DFT) of $f(x, y)$ is:

$$F(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \exp[-j2\pi(ux+vy)/N] \quad (4)$$

and the inverse transform is:

$$f(x, y) = \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u, v) \exp[+j2\pi(ux+vy)/N] \quad (5)$$

Figure 5 shows two sub-patterns and the correlation in the frequency domain from Eq 3: The peak indicates the value of the translation.

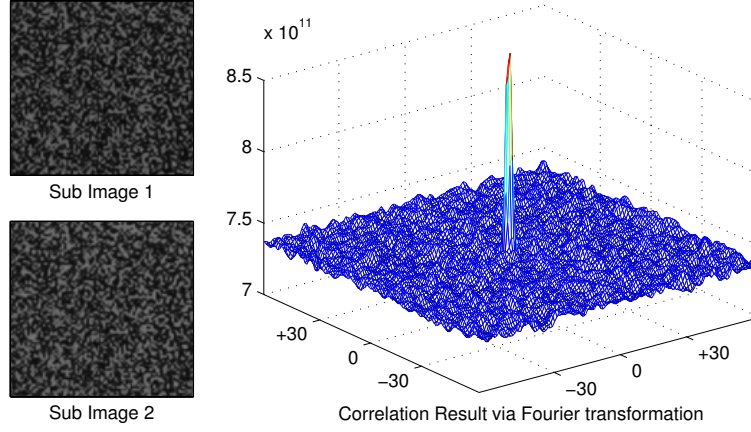


Figure 5: Displacement indicated by position of the peak in the correlation

In general, translations will be non-integral. To estimate a sub-pixel translation value, Chen et al suggested fitting a 2D parabola to the surrounding nine points [2]. Sjö Dahl increased the accuracy by shifting in the frequency domain until an autocorrelation occurs [6].

A 2D-parabola fit was used here: it is computationally simple and yielded consistent results with high frequency patterns. Figure 6 shows an example of x-direction displacements against positions in the image, using the patterns in Figure 5.

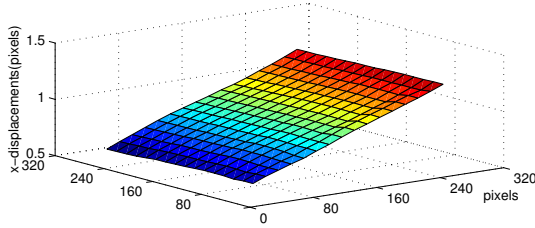


Figure 6: Example x-displacement plot

The random error in locating the correlation peak is [7]:

$$e = k\sigma^2 \sqrt{\frac{1-\delta}{\delta}} \quad (6)$$

where δ is the correlation factor, σ is the average speckle size, and k is a function of σ and the inverse of the window size. So, accuracy drops rapidly when correlation drops.

If there is a large displacement between two fields, the correlation function will have a smeared-out peak because of the decreased correlation (less overlapping area). In such cases, for a biggest possible overlapping area, the position of the window will be shifted to the new position until the integral translation is zero. The sub-pixel peak position is then estimated after this shift.

2.2 Deformation calculation

With calibrated stereo cameras, the position of a scene point can be obtained by intersecting two rays formed by the corresponding points in the left and right images. The process is outlined as follows.

After calibration, the intrinsic matrices K_i , rotation matrices R_i , and translation vector t_i of each camera are known. Their optical centres, O_1 and O_2 , are given in world coordinates as:

$$\begin{aligned} O_1 &= -R_1^T t_1 \\ O_2 &= -R_2^T t_2 \end{aligned} \quad (7)$$

Given a pair of corresponding image points, their 'ideal', undistorted coordinates $p_1 = (x_1, y_1, 1)^T$ and $p_2 = (x_2, y_2, 1)^T$, can be calculated from their real image coordinates providing the lens distortion parameters are known. The directions of the two rays, Δ_1 and Δ_2 , are defined as:

$$\begin{aligned} \Delta_1 &= P_1 - O_1 = R_1^T K_1^{-1} p_1 \\ \Delta_2 &= P_2 - O_2 = R_2^T K_2^{-1} p_2 \end{aligned} \quad (8)$$

Two rays do not always intersect in 3D space, so one may choose to approximate the intersection by finding the pair of points that have the shortest distance. Let the two rays be $O_1 + t_1 \Delta_1$ and $O_2 + t_2 \Delta_2$. The parameters, t_1 and t_2 , that give the shortest distance can be calculated by solving the following minimisation problem:

$$(t_1, t_2) = \arg \min_{t_1, t_2} (O_1 + t_1 \Delta_1 - O_2 - t_2 \Delta_2)^2 \quad (9)$$

Taking the partial derivatives with respect to t_1 and t_2 and setting them to zero, t_1 and t_2 are found by

$$\begin{pmatrix} t_1 \\ t_2 \end{pmatrix} = \begin{pmatrix} +\Delta_1^T \Delta_1 & -\Delta_1^T \Delta_2 \\ -\Delta_1^T \Delta_2 & +\Delta_2^T \Delta_2 \end{pmatrix}^{-1} \begin{pmatrix} -\Delta_1^T (O_1 - O_2) \\ +\Delta_2^T (O_1 - O_2) \end{pmatrix} \quad (10)$$

After finding t_1 and t_2 , the intersection, X , can be approximated as the midpoint of these two nearest points.

$$X = (O_1 + t_1 \Delta_1 + O_2 + t_2 \Delta_2) / 2 \quad (11)$$

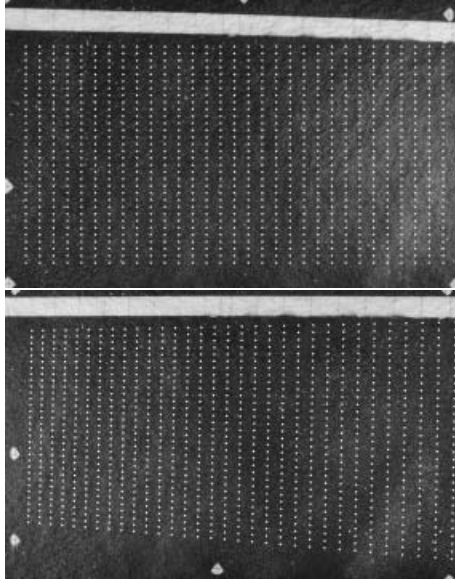


Figure 7: Initial correspondences established

2.3 Correspondence registration

Digital speckle photography records only the displacement vectors. For the triangulation to work, the initial coordinates in the left and right reference images must be established. Given a set of sample points in the left image, their corresponding points in the right image can be obtained by manual pairing or performing a conventional left-right stereo matching. However, manual registration is too time consuming if the number of sampling points is large and the left-right stereo matching suffers the aforementioned problems (listed in section 1) and did not always yield a robust result.

Since the experiment focuses on the relative depth change and the vacuum bag surface is nearly flat before the infusion, the initial bag surface can be regarded as a plane. Thus, correspondences at the initial stage can be approximated as a planar homography \mathbf{H} , a 3×3 matrix, induced by the bag surface:

$$x' = \mathbf{H}x \quad (12)$$

where x and x' are a pair of corresponding points in the left and right image respectively. Given coordinates of at least four pairs of corresponding points, the planar homography can be found by Direct Linear Transformation or other more elaborate methods [8].

If one chooses not to calculate the planar homography, an equivalent method is to back project the left point x to the bag surface plane (ray-plane intersection), and project it again into the second camera's image plane. Figure 7 shows an example of the initial correspondences established (white circular dots) via the initial surface plane.

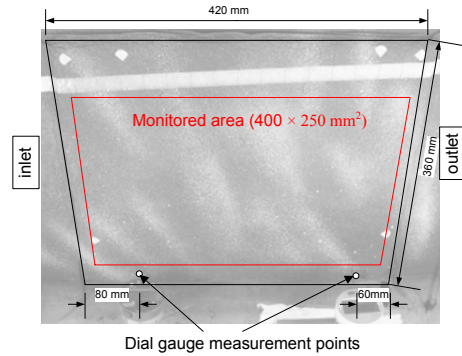


Figure 8: Layout of monitored area

3 Experimental results

In a set of evaluation experiments, infusion was simulated by oil instead of resin. Two convergent Canon EOS 20D cameras, separated by 800 mm, were mounted ~ 1 m above the infusion bag. Two dial gauges (Mitutoyo Digimatic 543-256, resolution 0.001 mm) were placed near the bag border at the inlet and outlet side. Figure 8 shows the layout.

A total of 26 image pairs were taken - images #0 to #21 were taken every 10 seconds, and images #22, #23, #24 and #25 were taken at 5, 10, 20, and 30 minutes after lling. The cameras operated at a resolution of 3504×2336 pixels and apertures were set to $f/22$ to maximise depth of field and minimise lens distortion.

The speckle pattern was prepared by applying a layer of undercoat (Dulux spraycote Flat Black) and then overcoated by spraying white paint (Dulux White Undercoat). These two layers were effective in reducing the reflective effect and provided a random speckle pattern - see Figure 5 (left).

The correlation window size was set to 64×64 pixels. A 64×64 window gave stable results with a reasonable computation time. To evaluate the stability of sub-pixel displacement estimation, for each point, displacements within a local 3×3 pixel window centred on it were calculated. Since the vacuum bag surface is smooth locally, all nine points should have similar displacements. Standard deviations of these displacements give a measure for sub-pixel estimation consistency: Table 1 shows standard deviations of displacements for 1024 evenly distributed points between the reference (#0) images and the #1 images.

	Left sequence		Right sequence	
	Average	Max	Average	Max
x-disp	0.0007	0.0033	0.0007	0.0037
y-disp	0.0007	0.0033	0.0007	0.0029

Table 1: Standard deviations of displacements within local 3×3 windows

Data obtained with the speckle photography system represented a complete description of the thickness

variations during processing. Figure 9 shows an example bag surface at 120s and Figure 10 shows thickness profiles at 20, 40, 60, 80 and 100 s after infusion began

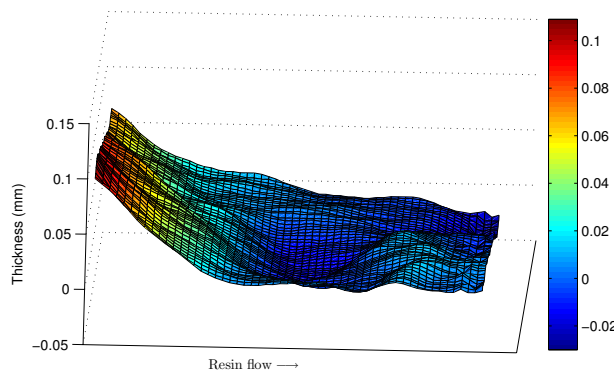


Figure 9: Bag surface (monitored area) at t=120s

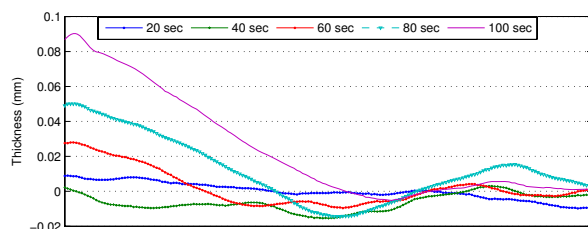


Figure 10: Thickness profiles at selected times

Figure 11 compares depths measured by the dial gauge at the outlet side and those calculated by speckle photography at the same horizontal position. The same trend was observed but speckle photography gave slightly higher values. One possible reason is that the dial gauge was near the bag border where there is lower variation than in the centre area. The dial gauge also requires contact and the application of some pressure. Thus it may be expected to underestimate an upward displacement.

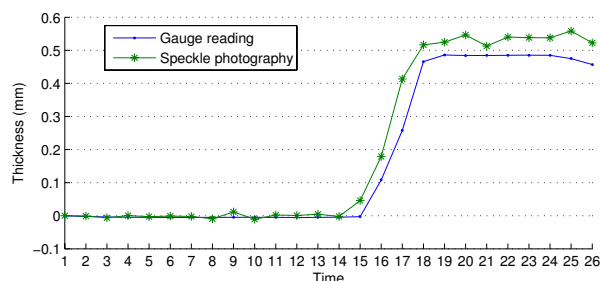


Figure 11: Speckle photography vs dial gauge readings

A reliable ground truth was not available, since a point cannot be simultaneously measured by speckle photography and other techniques: a dial gauge obscures the

measured point and a laser gauge provides lighting interference. Thus accuracy was inferred from displacement estimations. Standard deviations within small 3×3 pixel windows were ~ 0.001 pixels with a maximum less than 0.004 pixels. Assuming the estimation error to be within 2.5 standard deviations of the maximum standard deviation, an accuracy of 0.01 pixels in displacement estimation was obtained.

Due to the wide baseline, convergent geometry and high resolution cameras, the depth accuracy was ~ 0.3 mm per disparity. Hence, the depth accuracy was ~ 0.006 mm (in consideration of displacement errors in the left and right images).

4 Conclusion

Stereo photogrammetry is a non-contact technique that provides dense depth maps. Matching in two images taken from different viewing points is affected by several sources of noise, such as reflections and specular highlights, different optical or electronic gain settings, perspective distortion, etc. [9]. This makes sub-pixel estimates unreliable. Thus traditional stereo techniques could not measure the very small displacements over a wide area required for this application.

By tracing speckle patterns during deformation, the difficult left-right matching process migrates to correlation between images taken by the same camera. For a small patch, specular effects, perspective distortion and lens distortion are almost constant. Therefore, displacements can be closely approximated as pure translations of patterns and thus accurately calculated. Experiments with the system described here produced stable sub-pixel accuracy displacements. Standard deviations within small 3×3 pixel windows were ~ 0.001 pixels with a maximum standard deviation < 0.004 pixels. With such levels of displacement estimation accuracy, the depth resolution obtained was ~ 0.01 mm over an area of $\sim 0.1m^2$. The reconstructed thickness profiles and corroborating measurements with dial gauges confirmed that the speckle photography system produced such high accuracy reliably.

References

- [1] M. Sjö Dahl and L.R. Benckert, "Electronic speckle photography: Analysis of an algorithm giving the displacement with subpixel accuracy," *APPLIED OPTICS*, vol. 32, no. 13, pp. 2278–2284, May 1993.
- [2] D.J. Chen, F.P. Chiang, Y.S. Tan, and H.S. Don, "Digital speckle-displacement measurement using a complex spectrum method," *APPLIED OPTICS*, vol. 32, no. 11, pp. 1839–1849, April 1993.

- [3] P. Synnergren and M. Sjö Dahl, "A stereoscopic digital speckle photography system for 3-d displacement field measurements," *Optics and Lasers in Engineering*, vol. 31, no. 6, pp. 524–443, 1999.
- [4] A. K. Prasad and K. Jensen, "Scheimpflug stereocamera for particle image velocimetry in liquid flows," *APPLIED OPTICS*, vol. 34, no. 30, pp. 7092–7099, October 1995.
- [5] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, Addison-Wesley, Sept 1993.
- [6] M. Sjö Dahl, "Electronic speckle photography: increased accuracy by nonintegral pixel shifting," *APPLIED OPTICS*, vol. 33, no. 28, pp. 6667–6673, October 1994.
- [7] M. Sjö Dahl, "Accuracy in electronic speckle photography," *APPLIED OPTICS*, vol. 36, no. 13, pp. 2875–2885, May 1997.
- [8] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, second edition, 2000.
- [9] J. Liu, P. Delmas, G. Gimelfarb, and J. Morris, "Stereo reconstruction using an image noise model," *Digital Image Computing: Techniques and Applications (DICTA)*, pp. 69–76, 2005.

Interactive Hand-held 3D Scanning

R. J. Valkenburg, D.W. Penman, J.A. Schoonees, N.S. Alwesh, and G.T. Palmer

Industrial Research Limited, P.O. Box 2225, Auckland, New Zealand.

Email: r.valkenburg@irl.cri.nz

Abstract

This paper discusses some characteristics of free-form or mobile three-dimensional (3D) scanning. We demonstrate these through a description of a hand-held 3D scanning system for static objects and scenes ranging in size from less than a metre to tens of metres, indoors or outdoors. The scanner's pose is optically tracked relative to a constellation of active targets placed around the scene at the start of the survey. The system auto-calibrates the target locations and defines a scene coordinate system in which all scan data is subsequently represented. Mobile scanners can capture 3D structure of almost arbitrary complexity very rapidly. Real-time visual feedback to the operator coupled with manual control of data filtering can result in artefact-free 3D point clouds. Free-form scan data typically contains very few holes because the scanner can be manoeuvred to observe occluded surfaces, and oriented optimally for obtaining ranges to difficult surfaces.

Keywords: 3D scanning, pose tracking, range sensor

1 Introduction

Demand for three-dimensional (3D) computer models has risen dramatically with the advent of affordable processing power and graphics display capability. While good tools exist for creating synthetic digital models, applications that require the geometry (3D structure) of real-world scenes or objects are hampered by the cost and inconvenience of current 3D scanning technology:

- Objects with complex geometry have to be surveyed from many different viewpoints to avoid holes in the acquired surfaces.
- Fixed-station scanners rely on fiducials placed in the scene to register multiple scans to a common coordinate system. In some applications this can be a time-consuming and inaccurate process when many scans are needed to cover a complex object.
- Surfaces have to be sufficiently visible to the scanner to be acquired reliably.
- Some common scanner technologies generate artefacts (stray points) near edges, requiring interactive clean-up after the survey.
- Common 3D scanners typically scan both interesting and boring parts of the scene with uniform sampling density.
- Most current mobile or hand-held scanners fail to fully exploit their mobility due to their underlying pose tracking technology: magnetic sensors fail near ferromagnetic materials, optical sensors require unoccluded lines-of-sight,

inertial sensors drift, global positioning system (GPS) sensors only work outdoors, and so on.

This paper demonstrates the advantages of mobile scanning by reporting on a hand-held scanner designed to address most of these problems. Its first prototype is optimised for medium-sized scenes or objects (one to tens of metres) although this is not a hard constraint.

Free-form scanning is characterised by a highly interactive scanning experience: the operator sweeps the scene with a motion reminiscent of spray-painting, covering surfaces of high interest more densely, glossing over areas of lesser interest, and manoeuvring freely to scan into awkward regions such as concavities.

The system resembles other current hand-held or mobile scanners and differs from them in some key aspects. The Polhemus FastSCAN [1] uses magnetic sensors to determine the pose of the scanner which limits the working volume to be about a metre cube and free of nearby ferromagnetic materials. The 3rdTech HiBall [2] uses an array of light-emitting diodes (LEDs) overhead (usually ceiling-mounted) to track the pose of a hand-held sensor which can be fitted with a stylus for surface contour tracing. Hand-held range sensors are surveyed in [3].

The applications of mobile scanning overlap with those already being addressed by other types of scanning. They include:

- Forensics and accident scene recording
- Movie visual effects and computer games
- Heritage and archaeology
- Virtual tourism
- As-built engineering surveying

It is likely that, in each of these sectors, mobile scanners will have wider utility in cluttered or convoluted environments than other types of scanning.

Some applications may be practically infeasible without scanner mobility. It is quite difficult, for example, to scan the cabin interior of a passenger aircraft or luxury yacht without being able to move the scanner freely between seats, bulkheads and other obstacles.

The paper is organised as follows: Section 2 describes our hand-held scanner in terms of its system components. Section 3 generalises the description to a discussion of some characteristics of mobile scanning and how it addresses many problems that hamper fixed 3D scanners. Section 4 provides a practical illustration through an example scan. Section 5 suggests future work, specifically the potential for highly photorealistic renderings of 3D models captured by mobile scanners.

We use the phrases “mobile” and “free-form” interchangeably. They refer to the manoeuvrability associated with hand-held scanners, but may also apply to other deployments such as robot, vehicle or aircraft-mounted scanners.

2 System Description

The basic functions of the scanner system are shown in Figure 1. The function of each of the blocks will be described in the following sections.

2.1 Conventions

Figure 2 shows the relationship between some of the coordinate systems used in the system. For notational ease, the scene coordinate system is called CSW (world coordinate system), and the scanner head’s coordinate system is called CSM (mobile coordinate system). CSR is the range sensor’s coordinate system. The pose sensor’s coordinate system is denoted CSP, and CST represents the coordinate system of additional sensors such as texture sensors.

2.2 Base station

The base station computer is used for overall system control and for storage and visualisation of scan data.

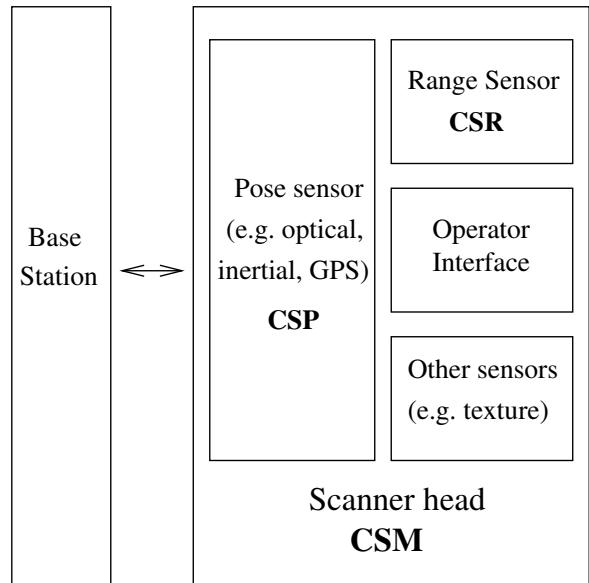


Figure 1: Functional block diagram.

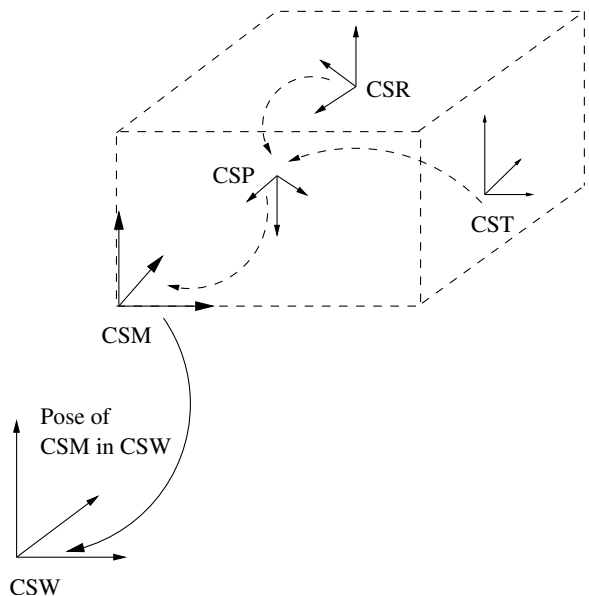


Figure 2: Coordinate systems and their relationships

2.3 Operator Interface

The operator interface comprises a display, indicator lights, and buttons on the scanner head. An important use of the display is to provide real-time visual feedback of the accumulating scan data. This is referred to as the *guidance visualisation*.

2.4 Pose Sensor

The pose sensor estimates the pose (location and orientation) of CSP in CSW in real time. Any spatial data produced by a *local sensor* rigidly attached to the scanner head can then be transformed into CSW.



Figure 3: The prototype scanner head, resting on its laser line scanner on which six direction sensor cameras and an operator panel are mounted.



Figure 4: Scanner prototype: On the table from left to right are the base station, active targets which would be placed around the scene before the survey, and the scanner head. The scanner head is connected by umbilical cord to a processor-filled backpack.

The scanner uses an optical tracking system comprising an omnidirectional camera on the scanner head and a number of active targets in or around the scene.

The omnidirectional camera is approximated in the prototype by six synchronised digital cameras, called *direction sensors*, with wide-angle lenses and optical filters matched to the colour of the targets (see Figure 3). The targets are segmented from each image and their centroids are extracted to sub-pixel accuracy. These centroids are then transformed into 3D lines in CSP. The lines together with the known target positions are used to estimate the pose of the scanner.

The geometric relationship between each of the direction sensors is fixed and calibrated (see section 2.7 below) so that the group can be characterised by a single coordinate system denoted by CSP.

2.5 Range Sensing

The scanner prototype uses a time-of-flight laser line (or flying-spot) range scanner. Since it is rigidly fixed to the scanner head, and its pose CSR in CSM is calibrated, the laser scan data can be represented in CSW as soon as the scanner head's pose is known (CSM in CSW), which can be done in real time.

The laser line scanner accuracy of about ± 10 mm (3 standard deviations) currently makes the biggest contribution to system accuracy. Different applications would typically require range scanners with different reaches, wavelengths and accuracies. For example, sub-millimetre accuracy would probably require a laser triangulation device with a reach of less than a metre.

Every data point is tagged with the pose of the scanner at the instant it was captured. The output is therefore more than a point cloud, it contains information on observation directions which can be used to disambiguate the sidedness of surfaces, especially those of thin sheets.

2.6 Data representation

All geometric primitives in the system are represented and communicated as homogeneous geometric algebra (GA) [4, 5] elements: points, lines, vectors and rotations. Work is in progress to port the system to conformal GA which is a more powerful and compact description [6]. A pose, for example, comprises a GA rotation and a GA vector in the homogeneous model, and a GA motor in the conformal model.

2.7 Calibrations

The system depends crucially on three levels of calibration, namely intrinsic, group, and target self-calibration. Intrinsic calibration relates to characterising sensor outputs (such as images from cameras) as meaningful measurements of the scene (such as angles between targets and camera optical axis). Group calibration relates to the relative poses between sensors and the scanner head. Target self-calibration relates to finding where the active targets are in the scene at the start of a survey, and establishes the scene coordinate frame.

2.7.1 Intrinsic and group calibrations

Intrinsic camera calibration [7] is performed for each of the direction sensors which make up the pose sensor. This effectively turns each direction sensor camera into an accurate meter of angles of

rays from the scene going through the image plane and optical centre.

Camera group calibration establishes the pose of each sensor in CSP. Similarly, laser range sensor group calibration finds the pose of the range sensor relative to the scanner head.

After intrinsic and group calibration the pixel coordinates of a target's centroid can be transformed to a line in CSM and represented internally as a GA element.

In the current implementation, the pose sensor coordinate system CSP coincides with CSM.

2.7.2 Target self-calibration

Target self-calibration is described in [8]. It involves moving the scanner to a number of stationary positions in the scene. The pose sensor gathers a set of lines in CSP at each position. These sets of lines, and a yardstick for scale, are presented to the target calibration algorithm which determines the 3D position of each target.

These target estimates can be further refined using an iterative algorithm [9] on additional target sightings gathered by walking around the scene.

3 Characteristics of Mobile Scanning

We believe that any efficient solution to the scanning of complex shapes has to involve a highly mobile scanner. There is currently a preponderance of statically-mounted scanners. The advantages of mobile scanning can be discussed one by one: in combination we think they provide a compelling argument for the use of mobile scanning in many practical situations.

Many of the advantages stem from the highly interactive nature of scanning and the possibility of real-time visual feedback of the growing 3D data set to the operator.

In our system scanning is activated by a trigger button and can be started, stopped and resumed at any time. A common surveying pattern seems to be one or several scanning sweeps followed by the operator moving to the next position before resuming the survey.

3.1 Data Quality

Holes in surface meshes obtained from scanners has been a persistent difficulty requiring post-survey intervention. Holes are caused by incomplete scanning due to occluded surfaces and failure of non-contact range sensors to detect surfaces, for example due to absorption of laser light.

The following characteristics of a mobile scanner help minimise holes in the data:

- The scanner head can follow complex trajectories to acquire data in areas that are difficult to access.
- The orientation of the scanner can be adjusted to be more normal to the surface in order to get data from surfaces with low return.

3.2 Discarding Unwanted Data

3D scanners can produce a lot of data, but quality or usefulness is not guaranteed. With real-time visual feedback to the operator, data filters can be made fully interactive. This effectively places operator intelligence into the filter cascade:

- If scan data looks anything but perfect, all or some of it can be discarded and immediately rescanned. The graphical operator interface of our prototype, for example, has a slider control which rolls back the survey in time, interactively showing the acquired point cloud at any earlier time, and allowing resumption of the survey from that time.
- If laser return intensity is considered too low to produce accurate ranging, those points can be discarded immediately. The operator reorients the scanner and rescans to get a stronger return.
- Stray points near edges and surfaces scanned at grazing angles can be identified by their sparseness and either discarded or visually tagged for the operator to make a snap decision on their fate.

The combined effect of such interactive filtering is that one has a clean set of data at the end of the survey, containing only the areas of interest thus minimising the post-processing.

3.3 Data Quantity

Handling very large data sets can be a problem. The above filters reduce data size by removing points which are suspect or simply unwanted. In addition, the following features allow us to control data quantity,

- Only areas of interest are scanned. For example, it is possible to scan discrete objects in a scene and maintain their spatial relationship without scanning the entire region in between.
- Areas can be scanned at suitable resolutions. It is often not necessary to scan the entire scene at high resolution.



Figure 5: The scanner prototype in action.

- With known pose of the scanner in CSW, it is trivial to define virtual bounding boxes containing the objects of interest. All points that are out of bounds are silently discarded. The bounding volume can be of any piecewise planar shape and may be formed interactively by indicating the bounding planes with the scanner. Bounding planes can be removed, and new ones added, interactively in the course of the survey.
- Bounds can also be placed on the range of acquired points from the scanner head. Points not within a specified window of ranges from CSM are automatically discarded in our prototype, making it easy to reject either foreground or background clutter. One might think of scanning an object in a barred cage without including any of the bars.

4 An Example Scan

Figure 6 shows the result of an example scan with the prototype scanner. The decapitated mannequin, simulating a forensic scene, spans about three metres. This scene is reasonably complex, including concavities, separated parts and undersides that are difficult to access.

The gap between the body and the head arises because the scan was intentionally restricted to those two areas of interest. The positions of the head and body are nevertheless spatially maintained to the accuracy of the scanner, even if they had been far more widely separated.

The floor appears bounded by a rectangle because a virtual bounding box, enclosing the volume of interest, was optionally defined at the start of the scan. No data was therefore collected of the surroundings.

The noisy points visible above the surface of the body is indicative of system accuracy. The scanner's accuracy, defined in terms of average or root-mean-square errors in scan points, is not easy to characterise simply. It is a non-linear function of many variables in several contributing subsystems: sensor intrinsic calibration, sensor group calibration, target self-calibration, pose estimation, and range sensor error among them.

In some cases accuracy can be measured as a function of residual error. Target self-calibration and pose estimation use lines between the direction sensors and each visible target. The root-mean-square angle between actual and reprojected lines can be found as a measure of either target self-calibration or pose estimation accuracy. How these numbers translate to eventual accuracy of 3D points again depends on factors such as the relative geometry of targets, scanner head, and scanned points.

System accuracy is dominated in the current prototype by the accuracy (or lack thereof) of the laser line scanner mounted on the scanner head. Future prototypes will employ more accurate range scanners.

The scanned objects have no significant holes that would make meshing difficult. It was straightforward to scan surfaces that would have presented a challenge to fixed-position scanners: around the legs and close to the floor.

5 Future Work

Future work could be based on the fact that handheld scanners can produce data sets in which each surface in the scene is seen from hundreds of directions. If a calibrated colour camera is attached to the scanner head, copious amounts of texture data can be gathered. The pose of the texture camera will be known for each texture element, which will in turn be closely registered to the underlying 3D geometry. Such texture data sets may be mined for detailed surface appearance models.

If, in addition, illumination of the scene is controlled or measured, bidirectional reflectance distribution function (BRDF) modelling of surfaces could yield highly photorealistic renderings with relighting [10].

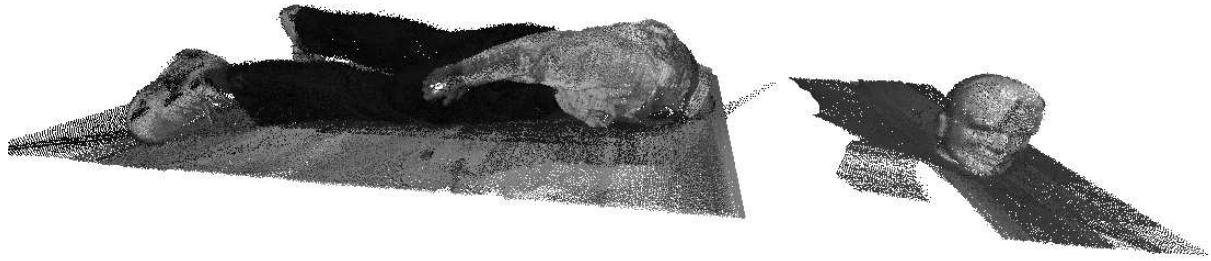


Figure 6: Captured point cloud rendered with laser return intensity.

6 Acknowledgements

This work was supported by the New Zealand Foundation for Research, Science and Technology.

References

- [1] J. Greco, “Polhemus FastSCAN,” *Cadence*, vol. 15, pp. 45–48, February 2000.
- [2] G. Welch *et al.*, “The HiBall tracker: High-performance wide-area tracking for virtual and augmented environments,” in *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST 99)*, (University College London), ACM, 20–22 December 1999.
- [3] P. Hebert, “A self-referenced hand-held range sensor,” in *Third International Conference on 3-D Digital Imaging and Modeling (3DIM '01)*, pp. 5–12, 2001.
- [4] W. E. Baylis, ed., *Clifford (Geometric) Algebras with Applications to Physics, Mathematics, and Engineering*. Boston: Birkhäuser, 1996.
- [5] C. Doran and A. Lasenby, *Geometric Algebra for Physicists*. United Kingdom: Cambridge University Press, 2003.
- [6] D. Tweed, “Estimating rigid motions via the conformal model of euclidean space,” in *17th International Conference on Pattern Recognition (ICPR'04)*, vol. 2, pp. 171–174, 2004.
- [7] R. J. Valkenburg and D. W. Penman, “Accurate unsupervised camera intrinsic calibration,” in *Image and Vision Computing New Zealand (IVCNZ 2004)* (D. Pairman, H. North, and S. McNeill, eds.), (Akaroa, New Zealand), pp. 215–219, 21–23 November 2004.
- [8] R. J. Valkenburg and N. S. Alwesh, “Calibration of target positions using the conformal model and geometric algebra,” in *Image and Vision Computing New Zealand (IVCNZ'05)*, pp. 241–246, Otago University, 2005.
- [9] R. J. Valkenburg, Y. Zhao, R. Klette, and N. S. Alwesh, “Iterative target calibration using conformal geometric algebra,” in *Image and Vision Computing New Zealand (IVCNZ'06)*, University of Auckland, 2006.
- [10] P. Debevec, “Virtual cinematography: Relighting through computation,” *Computer*, vol. 39, pp. 57–65, August 2006.

3D Visualisation Techniques for Multi-Layer Display™ Technology

Vijay Prema¹, Gary Roberts¹ and Burkhard Wünsche²

¹ Dept. Electrical & Electronic Engineering, University of Auckland, New Zealand.

² Department of Computer Science, University of Auckland, New Zealand.

Email: burkhard@cs.auckland.ac.nz

Abstract

Traditional computer monitors offer limited depth perception due to their 2D nature. The multi-layer display technology uses two or more display layers stacked in parallel and separated physically by depth. When viewing a Multi-Layer Display (MLD™) objects displayed on the front layer appear closer than objects on the back layer, and when moving the head while viewing the display objects on the front and back layer move relative to each other. However, it is not clear how complex 3D scenes can be rendered effectively using two physically separated view planes. We have experimentally analysed differences in perception when using single and Multi-Layer Displays and used the results to develop novel rendering techniques for MLD™. We found that perception of scenes can be improved by emphasizing important objects by displaying them on a different layer, by separating datasets on different layers, by extruding objects across layers, by transitioning objects smoothly between layers and by making use of the transparency of the front layer. As a result of our user studies we present a set of guidelines for the most effective use of Multi-Layer Display technology for rendering 3D scenes.

Keywords: Multi-layer displays, 3D displays, visual perception, human-computer interfaces, visualization

1 Introduction

Consumer level display technology has advanced dramatically with the advent of plasma and LCD displays. One important feature that has yet to reach the mainstream consumer is real 3D depth in images. There are many display technologies which can achieve varying levels of 3D depth, however most are expensive, inconvenient or have depth limitations. The Multi-Layer Display (MLD™) developed by PureDepth functions similarly to a conventional LCD monitor except that it features a second screen directly behind the transparent front screen. Images can be rendered on either of these two layers which are separated by a small space, conveying a limited amount of 3D depth [1].

3D display technology has a wide range of applications in entertainment, advertising, medicine, military and other fields. Examples include animated signs, video games, television, heads-up-displays and design visualizations [2]. 3D depth in display technology allows images to be interpreted faster, with more clarity and more realism. The MLD™ is one of the most accessible depth limited displays because of its compact size, low cost and compatibility with common PCs. Traditionally its dual layers are used as discrete surfaces for displaying overlaid information and highlighting objects by making them appear physically closer to the user.

In this paper we present a number of novel rendering techniques for harnessing the power of MLD™ technology and creating more effective visualizations. Section 2 summarises results about human depth perception. Section 3 introduces 3D display technologies and the PureDepth MLD™ technology. In section 4 we analyse differences in perception of single- and multi-layer displays and use the results in section 5 to develop more effective rendering techniques for a MLD™. Section 6 presents the summary of results obtained by performing user testing for our novel rendering methods. In section 7 we draw conclusions to our research and suggest directions for further studies.

2 Depth Perception

Human beings perceive depth using a combination of depth cues. *Psychological depth cues* are attributes of a physically flat image which are interpreted by the brain as 3D distance information and are hence extensively used when rendering 3D scenes on conventional 2D displays. *Linear perspective* is the recognition of parallel lines converging towards a point in the distance. Closer objects appear larger than distant ones and the known sizes of recognized objects can also be recalled from memory in order to make accurate distance estimations. *Occlusion* is the overlapping of objects and gives some idea of the order of objects in a scene. Depending on light sources, *shadows and shading* can provide clues as to

where objects are with respect to the ground plane. *Atmospheric perspective* is the blurring and blue tinting of distance objects due to scattering light. Some psychological depth cues involve motion. These include *motion parallax*, where nearer objects move faster than further ones; and *optic flow*, where the scene seems to expand from the point that the camera is moving towards [3][4].

Physical depth cues rely on the fact that humans have two eyes, and cannot be utilized by ordinary 2D displays. *Binocular disparity* is the main physical depth cue which involves the brain processing the images from both eyes. Since the eyes are some distance apart they capture slightly different images with a large overlap. The differences in the overlapping region can be perceived as 3D depth. *Vergence* is the movement of both eyes in opposite directions as they focus on an object which is moving towards the viewer. This can be used by the brain to very accurately judge distance [3][4].

The brain uses a weighted combination of all depth cues to perceive 3D depth. Physical cues are weighted more heavily at closer distances and psychological cues (particularly motion based cues) at long distances [3]. Gestalt psychology states that an important part of visual perception involves grouping parts of geometry in a scene into recognizable objects, e.g. by similarity, continuation, proximity, and common fate. Gestalt does not refer to depth perception in particular but we utilise the brains ability to perceive Gestalt when making objects appear to be continuous across both layers of the display [5].

3 3D Display Technologies

Artists have exploited size, shape, overlay, linear perspective and shadows to add depth to an image [6]. 3D displays are designed to utilise as many of the depth cues covered in section 2 as possible [7]. 3D capable technologies include anaglyphs, stereoscopic displays and autostereoscopic displays which use goggles or other tools to generate different images for both eyes [8]. All of these techniques suffer from user discomfort and eyestrain. A hologram records the intensity and the phase of the wavefront emanating from an objects surface but at present the images are fixed in film and cannot be manipulated. Volumetric displays illuminate points in 3D space but are very expensive [8].

Multi-Layer Displays do not have the same issues with discomfort, are smaller than other displays, can be easily installed on most computers (they require a dual head graphics card), and are cheaper than most alternatives. A Multi-Layer Display blends the colours of pixels rendered on the front and back layer together. This means if a dark pixel is rendered on the back layer, then the corresponding pixel on the front layer will also be dark. What is rendered on the front

layer must therefore take into account the colour of the scene behind it.

In our research we use a 17 inch MLD™ prototype, which consists of 2 LCD layers, with the back layer 7 mm behind the front layer. The display is connected to the computer via a dual head graphics card. The resolution of the screen is set to 2560x1024. The first 1280 pixels correspond to pixels on the front layer, the rest are for the back layer. We use OpenGL for rendering because it is platform independent, easily portable, offers fast real-time 3D graphics, has a stencil buffer for rendering silhouettes, and includes a shading language for implementing per-pixel operations. In OpenGL an easy way to render on the MLD™ is to create 2 viewports, one for the front layer and the other for the back, and render in each viewport separately.

4 Perceptual Differences for MLD™

We have performed and analysed a series of experiments in order to better understand how perception of the MLD™ varies from that of single layer displays (SLDs). Details of the experimental set-up and results are described in [9,10]. We found that users sitting within 0.5m from the screen in most cases could determine what was on the front layer and what was on the back layer. Reference objects helped with this which indicates that binocular disparity is an important depth cue in the MLD™. Similarly being able to move the head improved perception when using reference objects (motion parallax). Performance was further improved when the objects were overlapping.

5 Rendering Techniques on MLD™

We have developed various techniques to improve depth perception when rendering 3D scenes on a MLD™. The following subsections introduce these techniques and discuss their advantages, disadvantages and limitations.

5.1 Emphasising objects by putting them on a different layer

Since depth is more powerful than colour to help find an object [11], objects can be emphasized by putting them on a different layer, usually the front layer. Care must be taken when choosing colours for the emphasised object and the background scene. We found that the technique works best if the background has light colours, and the foreground has dark colours. If the background is dark then foreground objects are hard to see and if the foreground object is light it appears transparent (because of the physical makeup of the front layer) and the background shines through. The first problem can be alleviated by rendering a white silhouette of the foreground object onto the

back layer. We achieve this in OpenGL by drawing the background into the stencil buffer, then drawing the foreground objects in white where stencil values are non-zero, and finally drawing the foreground objects onto the front layer.

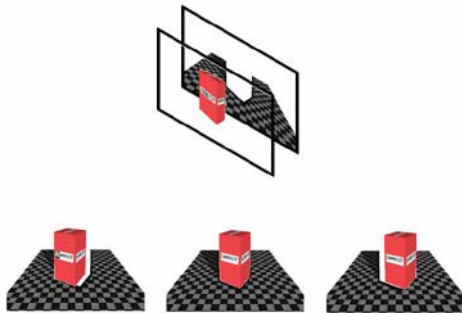


Figure 1: Emphasizing important objects.

We tested the scene displayed in figure 1 and found that most users perceived the red object as more accentuated when using the above described technique. One problem is that the white silhouette becomes visible when the user moves the head. This can be alleviated by fading the silhouette similar to the technique explained in the next subsection.

5.2 Determining layers by object depth value

The Z-value technique splits the entire scene by its Z-value (depth buffer value) and renders each half on a separate layer. All parts of the scene with a Z value greater than a certain threshold distance are rendered on the back layer and everything else on the front layer. An example is shown in the top row of figure 2, which shows a scene consisting of a rotating cube suspended in space and casting a shadow onto a platform below it. As the camera moves towards an object which is on the back layer, the object will eventually cross the threshold distance and gradually move to the front layer. The faces of any 3D object which intersect the threshold plane will be cut accordingly and the object will be partially rendered on both layers.

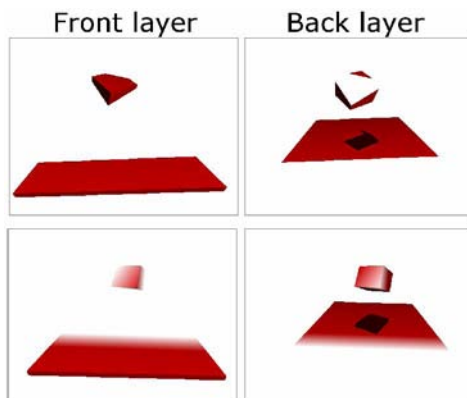


Figure 2: The Z-value technique using hard edges (top) and continuous shading (bottom).

The initial implementation of this technique was not very effective because of the discontinuity in the image caused by objects crossing the Z threshold. The main problem is that objects cut by the threshold plane appear to be unnaturally discontinuous or overlapping, particularly when the viewer moves the head.

The Z-value technique was dramatically improved by using *continuous shading* rather than discretely splitting a scene and rendering each half on a separate layer. In this implementation, each pixel in the scene is rendered with an independent alpha which depends on its Z-value and four other constants. These constants are $fMinZ$, $fMaxZ$, $bMinZ$ and $bMaxZ$. The alpha of each pixel is calculated using equation (1) for the front layer and equation (2) for the back layer.

$$fAlpha(z) = \begin{cases} 1 & z < fMinZ \\ 0 & z \geq fMaxZ \\ 1 - \frac{fMinZ - z}{fMinZ - fMaxZ} & otherwise \end{cases} \quad (1)$$

$$bAlpha(z) = \begin{cases} 0 & z < bMinZ \\ 1 & z \geq bMaxZ \\ \frac{bMinZ - z}{bMinZ - bMaxZ} & otherwise \end{cases} \quad (2)$$

The constants can be adjusted in order to provide enough overlap of the shading to naturally blend the layers. The demo application uses a custom OpenGL fragment shader to adjust the alpha for each pixel in real time [12]. The default values used are $fMinZ = 0.91$, $fMaxZ = 0.96$, $bMinZ = 0.88$ and $bMaxZ = 0.93$. Figure 2 shows a comparison of using hard edges on silhouettes compared to continuous smooth shading. Although this technique can be applied to any 3D scene with Z-values available, the amount of depth added to the scene is limited, as the distance between the layers is small. Another limitation is that the continuous smooth shading is less effective when rendering more complex objects with detailed surfaces (textures), particularly if the gradients overlap significantly.

5.3 Gradients

Simple view plane aligned static objects can be rendered effectively by splitting them and rendering the outer part on the front layer and the inner part on the back layer and fading the parts at the contour where they were split using the OpenGL smooth shade model.

We tested this technique by showing the scenes displayed in figure 3 and figure 4 to users. The three images at the bottom of each figure show the perceived scene from a view point to the left, in front and to the right of the monitor. The viewers were able to tell when the scene was rendered on only one layer

and all users agreed that using two layers improved depth perception. Further user studies showed that the technique is most effective when the width of ring object is small (the size of the two rings are around the same size) and the length of the gradient is a long. When the width of the ring is large, viewers can't see any difference from the equivalent single layer technique. The technique is only effective when the ring appears facing up. The most suitable background colour for ring area (which determines the colour of the highlight) is white or a colour lighter than the colour of the ring. When it's dark it makes the part rendered on the front layer hard to see. Possible explanations for these observations are that the ring appears less flat since when is rendered on both layers and that the whitish region where the rendered parts overlap moves as a viewer moves their head, which is consistent with how a specular reflection on a ring would behave when being viewed.

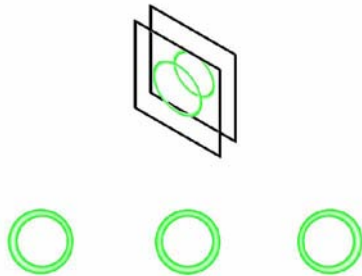


Figure 3: Effective use of gradients for ring objects.

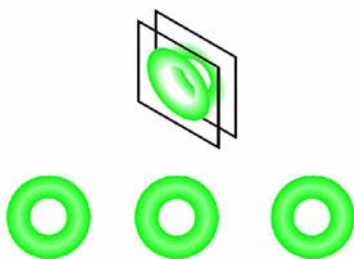


Figure 4: Ineffective use of gradient.



Figure 5: Ineffective use of gradient on other objects. We applied this technique to other objects such as the car in figure 5, but found it to be ineffective. The

most likely explanation is that the depth of the object is too large, i.e. when the viewer moves the head the headlights move independently of the rest of the car, which is unnatural. In addition the shape and behaviour of the whitish region where the scene components meet is inconsistent with that of a specular highlight.

5.4 Transitioning between layers

Objects can be made to appear between the display layers by rendering a percentage of the object on each layer. The object appears closer to the layer which has the higher percentage of the object rendered on it. We implemented this technique in OpenGL using alpha blending.

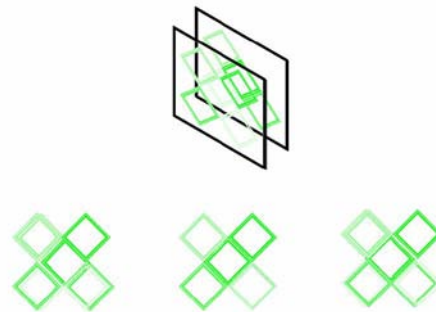


Figure 6: Transitioning objects between layers.

Two versions of the scene displayed in figure 6 were shown to users who were asked to order the squares by depth away from them. For the first version five out of six viewers agreed on the expected ordering, with the last viewer disagreeing in 3 positions. For the second version 3 out of six agreed on the expected ordering, with 2 disagreeing on 2 positions and one disagreeing on 3 positions. Note that is possible, that viewers use the apparent size of the gap between objects displayed on both layers to order the squares. However, overall the technique is effective for moving objects between layers. Possible problems are that objects change their perceived colour when moving between layers.

5.5 Calibrating objects to be viewed from a particular position

One way to render on the MLD™ is to assume that the viewer will only view the scene from a particular angle and to render the scene from that viewpoint such that both layers show the correct projection. However, it is very hard for the user to keep the head completely fixed and since we would lose depth perception due to motion parallax this technique does not seem suitable.

5.6 Grey scale depth map to determine layer

One of the most promising techniques we developed utilizes two images: the image to be displayed and a greyscale depth map which dictates how to render the

image on each layer. A white and black pixel in the depth map results in the corresponding image pixel to be displayed on the front and back layer, respectively. For gray scale pixels we blend the images between the layers by using the gray scale value as the weighting factor for alpha blending. Viewers found this technique to be very effective for an image of a brick wall (figure 7). The bricks appear closer and the grouting appears to recede behind the bricks.

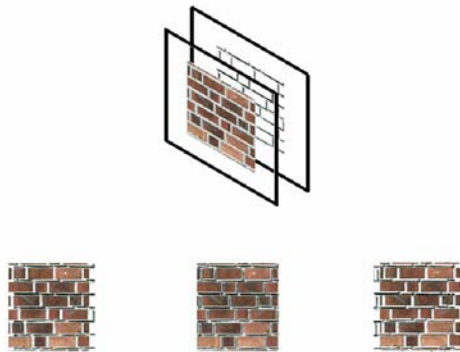


Figure 7: Depth Map to determine layer.

5.7 Visualising two data sets simultaneously separated by physical depth

Two data sets with matching domain (independent variable) can be effectively visualised on the MLD™ by displaying the data sets on a different layers on top of each other as illustrated in figure 8. The main advantage over a single layer display is that both datasets are physically close on the display which makes it easy to compare values for the same point in the domain. When points on one dataset block points on the other set the user can simply move the head to see the missing points.

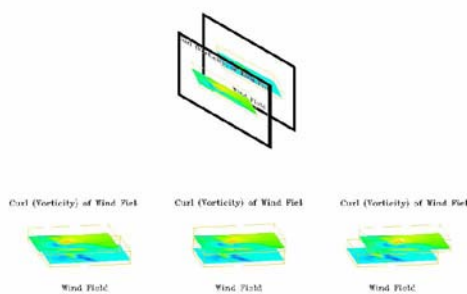


Figure 1: Comparing two datasets.

5.8 Moving objects on two layers with different speeds

Moving the scene on the front layer at a different rate to the scene displayed on the back layer gives the impression of a moving camera. An example is to have moving stars on the back layer and a stationary

spacecraft on the front layer. However, user testing indicates that the technique is equally effective for single layer displays.

5.9 Transparency

The transparent front layer can be utilised to display semi-transparent materials such as glass, water and fog. The scene rendered on the back layer then appears to be physically behind the semi-transparent material. This is in particular the case if the semi-transparent material is textured, e.g. slight waves, in which case motion parallax enhances depth perception. While this technique seems promising we did not have time to explore it in more detail.

6 Results

6.1 Analysis of Experimental Results

The perception of a MLD™ differs from a SLD in two ways. Firstly what is displayed on the front layer appears closer and separated from what is displayed on the back layer; this is due to binocular disparity. Secondly what is displayed on the front layer moves relative to what is displayed on the back layer when a viewer moves their head; this is due to motion parallax. Techniques that utilize either or both of these two properties to their advantage are more effective on the MLD™ than for a SLD.

For example, visualising two data sets simultaneously (figure 8) effectively makes use of both properties and works well. The gray scale depth map technique (figure 7) effectively makes use of binocular disparity and also works well. Techniques that don't make use of these two properties look identical on the MLD™ and SLD. Techniques where these depth cues interfere with the displaying data reduce the perceived information. An example is figure 5 which makes poor use of motion parallax and therefore appears confusing to the user.

6.2 Rules for creating effective 3D displays on MLD™

Our research found no general technique that works well for all applications. A developer must make intelligent decisions about what to render on the front and back layer to produce an effective scene. The following rules are compiled from our experiences will help to make this decision.

Emphasize important objects

Rendering a scene on the back layer and putting selected objects onto the front layer emphasises them. Other techniques such as simulating depth using gradients also accentuates objects. This is useful in applications such as advertising, where the advertised product can be accentuated, and visualization applications such as satellite information where the

designer wants to emphasise GIS information or military activity.

Making use of layer separation to separate information

Putting different datasets on different layers clearly shows that the datasets are separate but at the same time enables the user to read and compare both datasets.

Extruding objects across layers

Rendering an object over two layers, as explained in subsection 5.2 and 5.3, can give the illusion of physical depth and makes the scene more eye-catching.

Transition objects between layers

When moving objects between layers it is best to fade them between the two layers to give a continuous movement. This is useful when animating an object in 3D and a gradual movement between layers is required in order to emphasise its motion towards or away from the camera.

Making use of transparency

The transparency of the front layer can be used to render semi-transparent materials, such as glass, water and fog. The objects rendered on the back layer appear to be physically behind the semi-transparent material.

Avoiding visual discontinuity

When rendering a scene on the MLD™ it is important to take into account user head movements and that multiple users might view the display at the same time. In particular visual discontinuities as illustrated in figure 2 (top row) and figure 5, must be avoided.

7 Conclusion

Binocular disparity and motion parallax are the main depth cues users employ in order to determine which objects are displayed on the front and the back layer of the MLD™. Binocular disparity makes objects on the front layer appear closer and separated from what is displayed on the back layer. Motion parallax causes objects displayed on the front layer to move relative to objects displayed on the back layer when the viewer moves the head.

These depth cues cannot be depicted on SLDs and we have used them to develop effective rendering techniques for the MLD™. Gradients are useful for both reducing discontinuity caused by the physical gap between layers and for making objects appear continuous across layers. An effective general technique is to split a scene by Z-value to add a limited amount of physical depth to the scene. Important objects or objects that are closer to the viewer should be rendered on the front layer. In general, areas of an image can be made to appear

some distance between layers by rendering them with appropriate transparency values on both layers. The example with the brick wall in figure 7 demonstrated that this works best if only a relatively small depth is simulated.

Care must be taken that the physical separation between layers does not lead to unnatural effects such as gaps between layer images and unrealistic motion parallax (see figure 5 where the car's head lights move in an unnatural way).

In future research we want to develop an OpenGL style graphics library for use with MLD™. This might involve the development of special graphics card drivers to make full use of hardware acceleration.

References

- [1] PureDepth. *PureDepth Multi-layer Display*. Retrieved from <http://www.puredepth.com> on 15 March 2006.
- [2] R. Delaney, Forget the Funny Glasses. *IEEE Computer Graphics and Applications*, 25(3), May/June 2005, pp. 14-19, 2005.
- [3] G. Mather, *Foundations of Perception*. Psychology Press, USA, 2006.
- [4] D. McAllister, *Display Technology: Stereo & 3D Display Technologies*. Retrieved from <http://research.csc.ncsu.edu/stereographics/wiley.pdf> on 15 March 2006.
- [5] C. Pedroza, *Visual Perception: Gestalt Laws*. Retrieved from <http://coe.sdsu.edu/eet/Articles/visualperc1/start.htm> on 28 March 2006.
- [6] W. Ittelson, *Visual Space Perception*. Springer Publishing, USA, 1960.
- [7] T. Widjanarko, Brief survey on Three-Dimensional Displays: from Our eyes to Electronic Hologram, 2001. Retrieved 23 April 2006, from <http://filebox.vt.edu/users/twidjana/cv/holopaper.PDF>.
- [8] Sullivan, A (2004) 3-Deep: New displays render images you can almost reach out and touch. *IEEE Spectrum*, May 2004, p 30-35.
- [9] V. Prema, 3D Visualization Techniques with Multi-Layer Display Technology, SE Project Report, University of Auckland (submitted for publication).
- [10] G. Roberts, 3D Visualization Techniques with Multi-Layer Display Technology, SE Project Report, University of Auckland (submitted for publication).
- [11] Nakayama, K. & Silverman, G.H. (1986). Serial and parallel processing of visual feature conjunctions. *Nature*, 320, pp. 264-265.
- [12] Kessenich, J. (2006), *The OpenGL Shading Language*. Retrieved 1 September 2006 from <http://www.opengl.org/registry/specs/ARB/GLSLLangSpec.Full.1.20.6.pdf>.

SRICP: An Algorithm for Matching Semi-Rigid Three-Dimensional Surfaces

Ajmal Mian, Mohammed Bennamoun and Robyn Owens

School of Computer Science and Software Engineering

The University of Western Australia

35 Stirling Highway, Crawley, WA 6009, Australia

Email: ajmal@csse.uwa.edu.au

Abstract

The Iterative Closest Point (ICP) algorithm can accurately register and match 3D rigid surfaces. However, when it comes to non-rigid or semi-rigid surfaces such as the human face, the performance of ICP drops significantly. In this paper, we extend the ICP algorithm and propose a Semi-Rigid ICP algorithm which can match and register semi-rigid surfaces. We compare the performance of SRICP and ICP algorithms in a challenging scenario whereby 3D faces under exaggerated facial expressions are matched to 3D faces under neutral expression for recognition. Our results show that the proposed SRICP algorithm performs significantly better than the original ICP algorithm.

Keywords: Non-rigid registration and matching, semi-rigid surface, 3D face recognition, ICP.

1 Introduction

Surface registration and matching are fundamental problems in computer vision. Surface registration has applications in 3D modeling [8] whereas surface matching has applications in object recognition [11]. This paper mainly focuses on the latter problem i.e. surface matching with particular emphasis on 3D face recognition. A 3D face is essentially a three-dimensional surface represented by a 3D vector of its x, y, z coordinates. 3D face recognition is believed to have the potential to achieve higher accuracy compared to its 2D counterpart mainly because 3D surface matching is robust to changes in illumination, makeup and pose [3]. However, 3D face recognition is more sensitive to varying facial expressions compared to 2D face recognition. For literature review of existing 2D and 3D face recognition algorithms, the interested reader is referred the surveys of Zhao et al. [16] and Bowyer et al. [3].

The Iterative Closest Point (ICP) [2] is a classic algorithm used for the registration and matching of 3D rigid surfaces. It has been extensively used for 3D face recognition [5][7][9][10]. However, strictly speaking, face is a non-rigid object and varying facial expressions can significantly change the 3D surface of the face. This is why the performance of ICP significantly deteriorates under varying facial expressions. For example in [7], the 3D face recognition rate for neutral faces is 98% whereas it drops to 68% for smiling faces. The recogni-

tion rate drops significantly even though there are no exaggerated facial expressions. In our earlier work [10], we demonstrated that it is possible to achieve high 3D face recognition accuracy by using only partial regions of the face which are comparatively less sensitive to expressions. However, such regions not only vary between individuals but also vary between different facial expressions. To determine these precise regions for an individual requires many training images under all possible facial expressions which are usually not available in practical situations. Furthermore, this approach will also require the pre-classification of the expression type (e.g. smile, frown, anger, disgust) before performing recognition.

On the pretext that under different facial expressions, some regions of the 3D facial surface deform significantly lesser compared to others, we treat the face as a semi-rigid object in this paper. We extend the ICP algorithm and present SRICP (Semi-Rigid Iterative Closest Point) which can match and register semi-rigid 3D surfaces such as the human face. Like the ICP algorithm, SRICP is generic and can be applied to any 3D or even nD non-rigid datasets. Briefly, SRICP dynamically determines the points of the probe face which are less likely to have been affected by facial expressions and matches them to the corresponding points of the gallery face. These points are different for each match i.e. for each probe versus gallery face. Moreover, we also calculate a weighted distance error between the two faces by giving confidence weights

to different points (according to their sensitivity to facial expressions) of the gallery faces. We compare the recognition performance of the two algorithms using the FRGC (Face Recognition Grand Challenge) dataset [14] and demonstrate that SRICP outperforms the ICP algorithm.

This paper is organized as follows. Section 2 gives a brief description of the ICP algorithm. Section 3 gives details of the proposed SRICP algorithm and its differences from the ICP algorithm. Experimental setup and results are given in Section 4 and Section 5 respectively. Conclusions are given in Section 6.

2 Iterative Closest Point Algorithm

The ICP algorithm assumes that the two surfaces (to be matched or registered) are already coarsely registered or an initial set of correspondences have been identified between them either manually or automatically through a feature matching algorithm [8]. The distance between these corresponding points is then minimized by applying a rigid transformation to one of the surfaces (see Section 3 for details). Next, the ICP algorithm iteratively establishes correspondences between the closest points of the two surfaces and minimizes the distance between them by applying a rigid transformation to one of the surfaces. The iterations stop when the distance error reaches a saturation value and cannot be further decreased. The end effect of the algorithm is that the two surfaces are registered and the final distance error value is used as a similarity metric between the two surfaces. The more accurately the two surfaces resemble each other, the lower is the error. Note that this error value is also dependent upon how accurately the “closest points” of the two surfaces represent the correspondences between the two surfaces.

A number of modifications have been proposed to improve the registration performance of the ICP algorithm. These modifications are mainly targeted at improving the correspondence establishment which determines the final accuracy of the algorithm. Setting thresholds on the allowed distance between the closest points and the angular difference between their normals have improved the registration performance of the ICP algorithm. Establishing correspondences along the sensor viewing direction has been found to improve face recognition performance [10]. Since ICP is a computationally expensive algorithm, many efficient variants have also been proposed [15]. ICP has also been extended to non-rigid intensity based registration of 3D volumes [6].

3 Semi-Rigid ICP

SRICP mainly differs from the ICP algorithm in determining the eligible closest point correspondences and calculation of the distance error. In ICP, the closest pairs of points that are within a certain distance threshold are considered corresponding points. However, in SRICP, the closest pairs of points whose *weighted* distance is less than a threshold are considered corresponding points. Moreover, these weights also count towards the calculation of the final error in SRICP. The SRICP algorithm is described in detail below.

We use our automatic pose correction algorithm [10] for providing a initial registration of the faces for onward refinement by SRICP. All faces are pre-processed to remove spikes and noise and fill holes. Next each face is normalized with respect to pose and sampled on a uniform square grid [10]. The resultant faces are facing front with origin (coordinate [0 0 0]) at their nose tip.

Let $\mathbf{P} = [x_i, y_i, z_i]^T$ (where $i = 1 \dots n_P$) and $\mathbf{G} = [x_j, y_j, z_j]^T$ (where $j = 1 \dots n_G$) be the point cloud of a probe and a gallery face respectively. \mathbf{P} and \mathbf{G} are matrices of size $3 \times n_P$ and $3 \times n_G$ respectively. Let \mathbf{k} be a vector of size n_G whose elements represent the confidence in the respective point of the gallery face \mathbf{G} . \mathbf{k} is calculated as follows.

$$\hat{\mathbf{G}} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_j \\ y_j \end{bmatrix} \quad (1)$$

$$\mathbf{r} = \sqrt{\sum_{j=x,y} \hat{\mathbf{G}}.^2} \quad (2)$$

$$\mathbf{k} = \frac{r}{\max(\mathbf{r})\pi} \arccos(\hat{\mathbf{G}}_{x./\mathbf{r}}) \quad (3)$$

Eqn. 1 maps the gallery face to the xy-plane and rotates it by 90° so that the x-axis passes between the eyes. In Eqn. 2, \mathbf{r} is a vector of the distances of each point of $\hat{\mathbf{G}}$ from the origin i.e. nose tip. In Eqn. 2 and Eqn. 3, $.^2$ and $./$ stand for point wise square and divide respectively. $\hat{\mathbf{G}}.^2$ is equal to a matrix whose each element is equal to the square of the corresponding element in $\hat{\mathbf{G}}$. Similarly, $\hat{\mathbf{G}}_{x./\mathbf{r}}$ is equal to a vector whose elements are equal to ratio of the corresponding elements of the vectors $\hat{\mathbf{G}}_x$ and \mathbf{r} ($\hat{\mathbf{G}}_x$ is a vector of the x coordinates of the pointcloud $\hat{\mathbf{G}}$). Note that \mathbf{k} has a negative polarity i.e. lower values of \mathbf{k} mean higher confidence. The confidence values decrease with increasing distance from the nose tip and with increasing absolute angle from the x-axis as shown in Fig. 1. From Fig. 1, we can see that the upper part of the face has higher confidence whereas the lower part of the face (the mouth) has the lowest

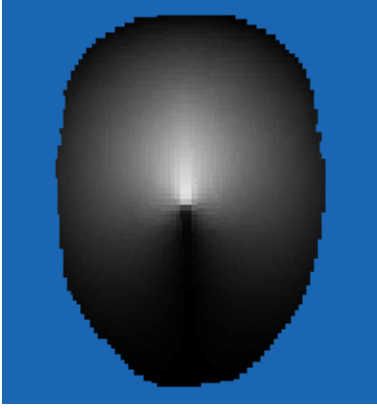


Figure 1: Confidence values calculated for a gallery face. The brighter regions represent higher confidence. Note that the upper part of the face and nearest to the nose has higher confidence.

confidence. This is to minimize the effects of an open mouth facial expression.

Let F be a function that finds the nearest point in \mathbf{P} to every point in \mathbf{G} .

$$(\mathbf{c}, \mathbf{d}) = F(\mathbf{G}, \mathbf{P}) \quad (4)$$

$$\mathbf{d}_k = \mathbf{d}(\mathbf{c})\mathbf{k} \quad (5)$$

In Eqn. 4, \mathbf{c} and \mathbf{d} are vectors of size n_P each such that c_i and d_i respectively contain the index number and distance of the nearest point of \mathbf{G} to the i th point of \mathbf{P} . \mathbf{d}_k is the confidence weighted distance calculated by multiplying the distance of a corresponding gallery point by its confidence (Eqn. 5). The correspondences are sorted according to the increasing value of \mathbf{d}_k and the last 10% are removed. Points of \mathbf{P} that fall in this category are also removed. Next, the 3D distance error e given by Eqn. 6 is minimized.

$$e = \frac{1}{N} \sum_{i=1}^N \|\mathbf{R}\mathbf{g}_i + \mathbf{t} - \mathbf{p}_i\| \quad (6)$$

In Eqn. 6, \mathbf{p}_i and \mathbf{g}_i are the corresponding points of the probe and gallery and $i = 1 \dots N$ (where N is number of remaining points of the probe). \mathbf{R} is a rotation matrix and \mathbf{t} is a translation vector that minimizes the distance between the corresponding points of the gallery and probe. Their values can be calculated using the classic SVD (Singular Value Decomposition) method [1]. Note, that this method can easily be generalized to any number of dimensions and is presented below for completeness. The mean of \mathbf{p}_i and \mathbf{g}_i is given by

$$\mu_p = \frac{1}{N} \sum_{i=1}^N \mathbf{p}_i \quad \text{and} \quad (7)$$

$$\mu_g = \frac{1}{N} \sum_{i=1}^N \mathbf{g}_i \quad \text{respectively.} \quad (8)$$

The cross correlation matrix \mathbf{K} between \mathbf{p}_i and \mathbf{g}_i is given by

$$\mathbf{K} = \frac{1}{N} \sum_{i=1}^N (\mathbf{g}_i - \mu_g)(\mathbf{p}_i - \mu_p)^\top \quad (9)$$

Performing a Singular Value Decomposition of \mathbf{K}

$$\mathbf{U}\mathbf{A}\mathbf{V}^\top = \mathbf{K} \quad (10)$$

gives us two orthogonal matrices \mathbf{U} , \mathbf{V} and a diagonal matrix \mathbf{A} . The rotation matrix \mathbf{R} can be calculated from the orthogonal matrices as

$$\mathbf{R} = \mathbf{V}\mathbf{U}^\top, \quad (11)$$

whereas the translation vector \mathbf{t} can be calculated as

$$\mathbf{t} = \mu_p - \mathbf{R}\mu_g \quad (12)$$

\mathbf{R} is a polar projection of \mathbf{K} . If $\det(\mathbf{R}) = -1$, this implies a reflection of the face in which case \mathbf{R} is calculated using Eqn. 13.

$$\mathbf{R} = \mathbf{V} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(\mathbf{U}\mathbf{V}^\top) \end{bmatrix} \mathbf{U}^\top \quad (13)$$

The above steps (from Eqn. 4 onwards) are iteratively repeated until the number of remaining points in \mathbf{P} are half the starting value i.e. $N \leq 0.5n_p$. The final value of error e_f between the two faces is calculated as

$$e_f = \sum_{i=1}^N (\|\mathbf{R}\mathbf{g}_i + \mathbf{t} - \mathbf{p}_i\| + \|\mathbf{R}\mathbf{g}_i + \mathbf{t} - \mathbf{p}_i\|(1 - k_i) + \frac{k_i}{2}) \quad (14)$$

All three terms in Eqn. 14 are first normalized on a scale of 0 to 1 for each recognition trial i.e. matching a single probe to all the gallery faces. This removes the bias between the terms. Moreover, the last term (k) is given half the weight of the other two terms.

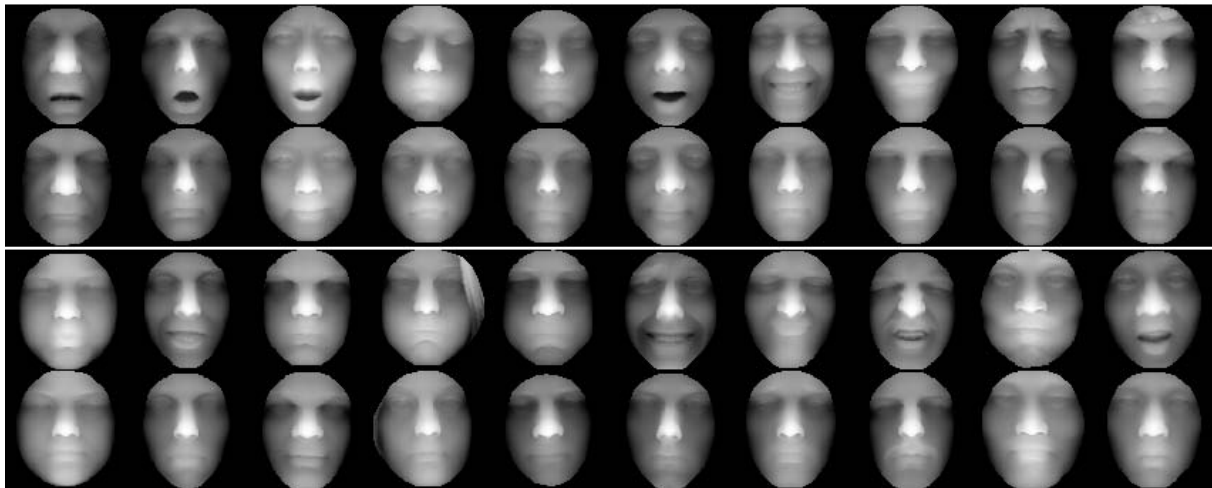


Figure 2: Sample 3D faces rendered as range images. First and third row contains the probe faces and second and last row contains their respective gallery faces.

4 3D Face Data and Experimental Setup

We used the FRGC v2.0 [14] data for our experiments which contains 3D faces along with their texture maps acquired with the Minolta Vivid 910 scanner [12]. However, we only used the 3D data for matching in this paper. There are 466 identities in the FRGC validation set. Out of these, only 374 individuals have a 3D face under neutral as well as non-neutral expression. For each of these 374 individuals, we selected one 3D face under neutral expression to form the gallery and one 3D face under exaggerated expression to form the probe. Selection was performed manually to ensure only those faces are selected which are significantly deformed due to facial expression e.g. blown cheeks and open mouth. Fig. 2 shows some example probes and their corresponding gallery faces. Moreover, where the choice of probe was to be made between a face that was covered with hair and a face that was not covered with hair, the former was selected to make the recognition extremely challenging.

Note that the aim of this paper is to compare the performance of the proposed SRICP algorithm to ICP under non-rigid (or semi-rigid) deformations using the same dataset. This is why we have only selected 3D probe faces under extreme facial expressions. We did not include the faces with neutral or minor expressions so that the results reflect matching performance on semi-rigid surfaces only. It is not the aim of this paper to achieve high recognition accuracy on the FRGC v2.0 data. Therefore, once the faces are preprocessed as described in [10], their alternate rows and columns are removed (i.e. the resolution is reduced by a factor of 4) in order to gain efficiency.

5 Results

We matched each probe to all the faces in the gallery once using the ICP algorithm and a second time using the SRICP algorithm. Fig. 3 shows the rank identification results. For each identification trial, the gallery faces are ranked according to their error scores e_f (Eqn. 14). A rank x recognition rate means the rate at which the correct identity is among the top x ranked identities. A rank one identification rate is the number of probes that were correctly identified (to its correct identity in the gallery) divided by the total number tested probes.

Fig. 4 shows our verification results. Each time a probe is matched with its correct identity in the gallery, the value of e_f is treated as a genuine score. However, when a probe is matched with a different identity in the gallery, the value of e_f is treated as an impostor score. The ROC curves are plotted as follows. The threshold for accepting a probe as a genuine client is varied and at every threshold, the verification rate is calculated as the number of genuine probes that fall below the threshold divided by the total number of genuine probes. Similarly, at every threshold, the False Acceptance Rate (FAR) is calculated as the number of impostors that fall below the threshold divided by the total number of impostors. At 0.001 FAR, SRICP achieves 62.57% verification rate whereas the ICP algorithm achieves 47.06% verification rate. These results show that SRICP outperforms the ICP algorithm.

The aim of these experiments was to perform an unbiased comparison of the SRICP and ICP algorithms on a challenging dataset. Note that the performance of both algorithms is quite low since

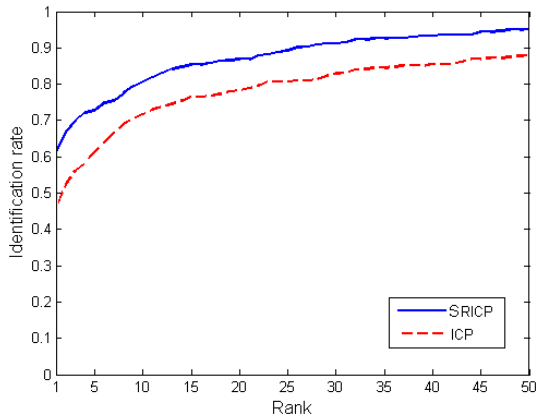


Figure 3: Identification results. SRICP achieves 61.23% rank one identification rate whereas the ICP algorithm achieves 45.99% rank one identification rate.

the data on which these experiments were performed was highly challenging due to exaggerated facial expressions. Moreover, the resolution of the faces was reduced (by a factor of 4) to gain computational efficiency. Therefore, these results can not be compared to others which used the entire FRGC v2.0 database (e.g. [10] and [13]) since the database also contains easy to recognize faces (i.e. with neutral and minor facial expressions). Moreover, these results can not be compared to those of Bronstein et al. [4] since their approach apparently does not deal with open mouth expressions.

6 Conclusion

We presented an algorithm for the registration and matching of semi-rigid 3D surfaces and demonstrated its performance on the challenging case of 3D face recognition under exaggerated facial expressions. Even though SRICP uses only half of the probe points for matching, it performs significantly better than the ICP algorithm which uses all the probe points from matching. SRICP is a generic algorithm and can be used for matching any 3D semi-rigid data. Moreover, like the ICP algorithm, it can also be extended to the nD case.

7 Acknowledgments

We would like to thank the organizers of FRGC [14] for providing the face data. This research is funded by an Australian Research Council discovery grant number DP0664228.

References

[1] K. Arun, T. Huang and S. Blostein, “Least-squares Fitting of Two 3-D Point Sets”, *IEEE TPAMI*, vol. 9(5), pp. 698–700, 1987.

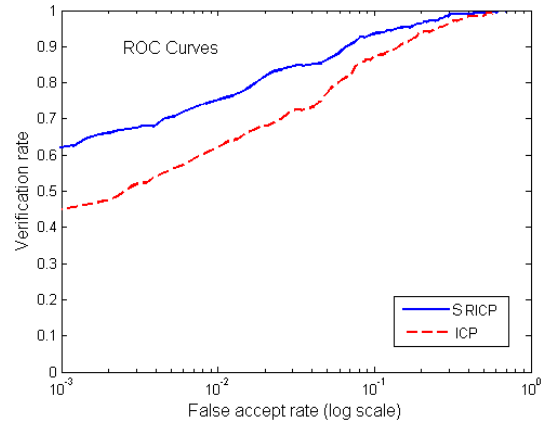


Figure 4: Verification results. At 0.001 FAR, SRICP achieves 62.57% verification rate whereas the ICP algorithm achieves 47.06% verification rate.

- [2] P. J. Besl and N. D. McKay, “Reconstruction of Real-world Objects via Simultaneous Registration and Robust Combination of Multiple Range Images,” *IEEE TPAMI*, vol. 14(2), pp. 239–256, 1992.
- [3] K. W. Bowyer, K. Chang and P. Flynn, “A Survey Of Approaches and Challenges in 3D and Multi-modal 3D + 2D Face Recognition,” *CVIU*, vol. 101, pp. 1–15, 2006.
- [4] A. M. Bronstein, M. M. Bronstein and R. Kimmel, “Three-Dimensional Face Recognition,” *IJCV*, vol. 64(1), pp. 5–30, 2005.
- [5] K. I. Chang, K. W. Bowyer and P. J. Flynn, “Multiple Nose Region Matching for 3D Face Recognition under Varying Expression”, *IEEE TPAMI*, vol. 28(10), pp. 1695–1670, 2006.
- [6] J. Feldmar, B. Malandain, J. Geclerk, N. Ayache, “Extension of the ICP Algorithm to Non-Rigid Intensity-Based Registration”, *Workshop on Mathematical Methods in Biomedical Image Analysis*, 1996.
- [7] X. Lu, A. K. Jain and D. Colbry, “Matching 2.5D Scans to 3D Models,” *IEEE TPAMI*, Vol. 28(1), pp. 31-43, 2006.
- [8] A. S. Mian, M. Bennamoun and R. A. Owens, “A Novel Representation and Feature Matching Algorithm for Automatic Pairwise Registration of Range Images”, *IJCV*, vol. 66, pp. 19–40, 2006.
- [9] A. S. Mian, M. Bennamoun and R. A. Owens, “2D and 3D Multimodal Hybrid Face Recognition”, *ECCV*, part III, pp. 344–355, 2006.

- [10] A. S. Mian, M. Bennamoun and R. A. Owens, “Automatic 3D Face Detection, Normalization and Recognition”, *3DPVT*, 2006.
- [11] A. S. Mian, M. Bennamoun and R. A. Owens, “3D Model-based Object Recognition and Segmentation in Cluttered Scenes”, *IEEE TPAMI*, vol. 28(10), pp. 1584–1601, 2006.
- [12] Minolta 3D Digitizers, “Non-contact 3D Laser Scanner”, <http://www.minolta3d.com>, 2006.
- [13] G. Passalis, I. Kakadiaris, T. Theoharis, G. Tederici and N. Murtaza, “Evaluation of 3D Face Recognition in the Presence of Facial Expressions: An Annotated Deformable Model Approach”, *IEEE Workshop on FRGC Experiments*, 2005.
- [14] P. J. Phillips, P. J. Flynn, T. Scruggs, K. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min and W. Worek, “Overview of the Face Recognition Grand Challenge”, *IEEE CVPR*, 2005.
- [15] S. Rusinkiewicz and M. Levoy, “Efficient Variants of the ICP Algorithm,” *3DIM*, pp. 145–152, 2001.
- [16] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, “Face Recognition: A Literature Survey”, *ACM Computing Survey*, pp. 399-458, 2003.

Objective Colour Measurement of Tomatoes and Limes

H.M.W. Bunnik¹, D.G. Bailey² and A.J. Mawson³

¹Farm Technology Group, Wageningen University and Research Centre, Netherlands

²Institute of Information Sciences and Technology, Massey University, Palmerston North, NZ

³Institute of Food, Nutrition and Human Health, Massey University, Palmerston North, NZ

Email: D.G.Bailey@massey.ac.nz

Abstract

Colour is an important parameter of fruit from which much information regarding the quality of the fruit can be gathered. Therefore a correct grading of fruit in research is necessary. Commonly only small areas of an object are processed, after which a rather coarse grading is applied. The aim of this study is to develop a low cost, objective technique that will measure the colour of limes and tomatoes. This system quantifies the colour of an entire object using four index numbers: the mean, standard deviation, skew and kurtosis, allowing an easy comparison between different objects or of the same object at different times.

Keywords: Colour change, tomato, lime, grading, image processing

1 Introduction

Colour is an important characteristic of fruit and vegetables. Consumers prefer bright red tomatoes over green or dark red tomatoes [1, 2]. Similarly their preference is for green limes, rather than yellow, since the green colour is linked to freshness [3]. Similar patterns can be found with many other fruits and vegetables.

Colour provides more information than just the marketability of the product. The colour of a tomato has a direct relation with the firmness [4]. Furthermore, the red colour gives a clear indication of the amount of lycopene (a carotenoid) in the tomato [5-7]. Lycopene is believed to have a preventive effect against several forms of cancer [8-10].

The importance of the colour of fruit is clear, yet there are many colour related questions still unanswered. To solve some of these questions, more knowledge is needed about the development of pigments such as chlorophyll and carotenoids as a function of time.

In colour research on fruit and vegetables it is common to classify the colour of an object into a number of bands. For instance a colour score has been given for limes by assigning a value of 0, 25, 50, 75 or 100% yellow [11]. The number of bands used can vary strongly from fruit to fruit. Tomato grading charts from auctions vary from 6 bands in the USA [12] to up to 12 classes in Israel and the Netherlands [2, 13, 14]. The majority of this grading work is done by expert panels and may result in errors [2]. Consumers in general have difficulty discriminating between adjacent colour grades in the most expanded colour charts [14].

In some cases a spectrometer is used with which a number of points upon the object are compared [1, 4, 11, 15]. The disadvantage of this method is that only a

small number of points on an object are measured, which can give a misleading representation of the colour.

To obtain a more objective manner of grading the entire object, an automated grading mechanism is needed. Research has been done applying spectral image analysis, with very good results [16-18]. The disadvantage of this method is that the equipment involved is relatively expensive.

The aim of this research is therefore the development of a relatively inexpensive, automated method of objectively grading an object in its entirety. This results in a set of numbers that describe the measured object in such a way that an easy comparison is possible between objects, or of a single object over time. The method is then used to quantify the colour change ('degreening') of limes and tomatoes over time.

This leads to the following research questions:

- Is it possible to follow the degreening of the complete area of a lime or tomato, using a standard video camera?
- How can an object be characterised by a small set of numbers that can be easily compared?

2 Instrument Design

2.1 Image Capture

To capture the images of the object (lime or tomato), a charge coupled device (CCD) camera is used (Sony DFW-SX900, resolution 960 x 1280 pixels). The camera is linked by firewire to a PC on which the processing software is installed.

To capture more than just one side of the object using a single camera, a turntable is used. The object is placed within a cup, to allow it to stand erect. The

turntable is driven by a stepper motor. To remove a stick-slip effect at low speeds, where the fruit rotates at a different speed to that of the cup, the inside of the cup is lined with rubber, which has a high coefficient of friction.

In the case of tomatoes, the inside of the cup is also lined with soft tissue, to prevent the edges of the cup from damaging the tomato. When working with limes this precaution is not needed, since the rind of a lime is less sensitive.

Eleven images are captured over one complete rotation, which takes 18 seconds. Each area is recorded several times, at different angles, but in one rotation the whole object is equally covered. Applying a higher rotation speed will re-introduce the stick-slip effect. Further using fewer images will increase fluctuations in the outcome.

2.2 Lighting

Even, consistent lighting is essential to obtain images of the object that are of a high enough quality. A DC halogen lamp avoids problems of mains flicker. It is also small, enabling a compact construction.

To obtain accurate measurements of the colour, specular reflections from the surface of the object must be avoided. This may be achieved using indirect, diffuse illumination. For this purpose the halogen lamp is fitted into a tube. At 20 cm. below the bulb a circular plate is fitted in the middle of the tube (see Figure 1). This prevents the direct illumination of the object. The inside of the tube is bright white, to scatter the light around the disk. The area below and the cup itself are also white. This allows light to be scattered back onto the object to provide relatively even indirect lighting.

The lighting obtained is good, but as can be seen in Figure 2, there is still a gradation in the white background, with the lower part of the image receiving less light than the upper part. Improving the light further, by for instance by using a white sphere rather than a cylinder, would improve the outcome.

At the bottom of the tube a window is created through which the camera obtains the images.

2.3 Algorithm

To process the images the program VIPS (Visual Image Processing System) is used [19].

When considering the colour change of limes and tomatoes, they both start as green objects. Whereas the tomato progresses until it is red, the lime stops changing 'halfway', when it is yellow. This means that a similar algorithm can be used for both cases.

Although the halogen lamp performs well with respect to illumination, the colour temperature of the lamp does not match the camera. As the inside of the

tube is white, this can be used as a reference for colour correction. The brightest 20% of the image contains only the background, so the average RGB value of these pixels provides an estimate of the white level. The colour correction requires that the background pixels are not saturated in any of the channels, as this would distort the average obtained. The black level is estimated empirically. A linear expansion is applied to each of the RGB channels to set the black level to 0, and the white level to 255. The result of this is shown in Figure 2.

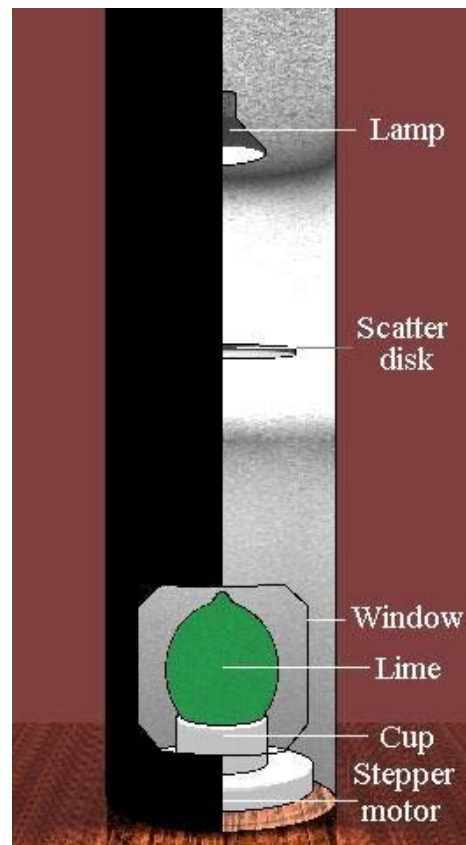


Figure 1: A sketch of the measuring device. The scatter disk ensures that only indirect lighting reaches the object.



Figure 2: Left before and right after applying the white balance and linear expansion.

The next step is to determine which pixels belong to the object being imaged. The blue channel stays approximately constant during the 'degreening' and is

of a similar low level for both limes and tomatoes. Therefore this channel can be used to automatically distinguish between the background and foreground. To enable a global threshold to be used, the blue channel is normalised by dividing by the maximum of the green and red channels. For limes, the green channel is relatively uniform and consistent, although for tomatoes, the green is replaced by red as the tomato ripens.

Finally, morphological filtering (using both opening and closing) removes small isolated areas that have been mis-thresholded to give a clear silhouette of the object. A small safety margin is removed from around the edges of the mask to remove the pixels near the boundary of the object which are viewed at a very acute angle.

Physiologically, the green channel strongly reflects the chlorophyll content of the object, whereas the red channel indicates the yellowness (or carotenoid content) of limes, or the lycopene content of tomatoes. Dividing the red by the green gives a usable numerical ratio.

$$pixel_colour_index_0 = \frac{Red}{Green} \quad (1)$$

Taking the ratio in this way overcomes the small unevenness in illumination as both the red and green channels will be affected equally.

While this index is useful for limes, where the yellow contains a strong green component, with tomatoes, the green component becomes much less than the red, and a small change in colour results in a large change of ratio. A more uniform colour index may be obtained from equation 2.

$$pixel_colour_index_1 = \frac{Red - Green}{Red + Green} \quad (2)$$

Again this index is normalised against variations in illumination by taking a ratio. This index ranges from -1 for green pixels, to 0 for yellow, and +1 for red pixels. The colour index values can be easily converted to a percentage by adding 1 and scaling. To obtain the index of just the object pixels, the colour index image is masked to remove the background.

2.4 Derived Index Numbers

Once the pixel colour index of the object is known, this has to be captured in a set of meaningful numbers. A histogram is obtained of the pixel colour index values accumulated over all 11 images, see Figure 3. In this way the histogram represents the complete surface of the object apart from a small region near the stem which is sitting in the cup, and the small region near the top that is always viewed acutely, see Figure 2. This area is in general less than 5 % of the total area.

From these histograms four index numbers are generated:

- Mean (μ)
- Standard deviation (σ)
- Skew (S)
- Kurtosis (K)

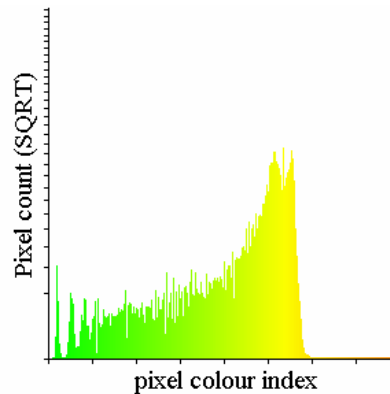


Figure 3: The histogram of a lime. Pseudo colours are applied relative to the colour of the lime.

It has been found that these four index numbers provide a good representation of the colour and colour distribution. The mean gives the average colour of the object. The standard deviation gives a measure of the range over which the colours are found. The skewness describes the symmetry in the distribution of colour. The kurtosis indicates the uniformity of the colour. These make it possible to easily compare different objects, or the same object over time.

Due to the large amount of data produced when making the measurements over a large number of objects and over an extended time, there is need for an automated processing system. This program is written in Visual Basic Excel and allows the input of large quantities of data in random order, producing the sorted data in a number of charts.

3 Initial Testing

Once the algorithm is working, the reliability of the entire setup is evaluated. For this purpose a number of tests are performed.

3.1 Consistency Testing

First the stability of the process is investigated calculating the index numbers over a long period (1 hour). In this time 100 separate measurements are made of the object, during which time the index numbers should remain constant.

The changes found in the measurements over this longer period are small and fall well within the margin compared to the coarseness of manual grading; for example see Figure 4 for a lime. During the one hour of measurement the index numbers show a stable outcome. The fluctuations stay within a range of less than 0.75 %.

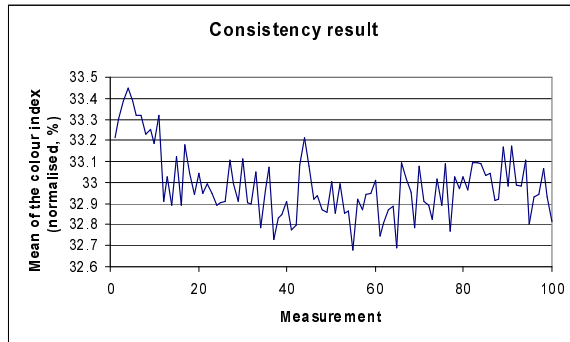


Figure 4: Fluctuations of the average colour index, during time. The y-axis is normalised between 0 and 100%. The fluctuation stays within a range of 0.75.

The other indexes showed similar, relatively small variations, although the kurtosis was sensitive to small variations in the low amplitude tails of the histogram.

The reason behind the small fluctuations is not known for certain. There is a small shadow between the cup and the object (visible in Figure 2) and this may introduce some uncertainty. With the rotation of the turntable, the fruit may also be moving slightly with the result that slightly different surfaces may be measured each time.

3.2 Ambient Lighting

The influence of ambient light sources, such as sunlight coming in the window, or the normal fluorescent lights in the laboratory, is determined. This is done by comparing index numbers under different light circumstances, while using the same object. Table 1 gives an overview of the effects.

Table 1: Effects of ambient lighting

Type of light	Mean	St. Dev.
Dark	55.15	0.006
Fluorescent	55.16	0.04
Sunlight	55.22	0.05

The effects of ambient lighting are negligible, as expected, due to the design of the measurement apparatus. The cylinder prevents any significant level of ambient light from reaching the object, apart from light coming in via the viewing window. The white balancing procedure corrects for the little light that does come in.

4 Measurement Performance

4.1 Test Setup

Finally, a small number of fruit (6 – 10) were measured, recording their colour change over time. To speed up the degreening process, the objects are kept at 20°C.

The tomatoes used are selected in a broad spectrum of colours. One was close to the breaker stage (<10%

coloured, whereas others were already close to red (>90% coloured) [12].

The limes used were approximately 70% green, but with quite large differences in colour uniformity.

4.2 Results

The initial test with a small number of tomatoes showed to be very valuable.

Figures 5-8 show the evolution of the measured colour index with time. The mean colour index clearly shows a development similar to that reported in the literature [6, 20]. Due to the partial character of this initial test, the curve does not show the complete progression of colour change.

The standard deviation increases slowly, after which it stays constant at a rather high level. This is due to the relatively large, even tails in the histogram. This clearly shows the benefits of the kurtosis, since this gives a clear indication about the ‘peakness’ of the histogram.

For the tomato marked with ‘+’ the measurement of the skew is different from the others. The tomato involved was initially very green (almost breaker). Due to the green colour, with a small red component, the skew is negative. In all other cases the colour is more uniform or with more red than green, resulting in a skew close to zero or positive respectively.

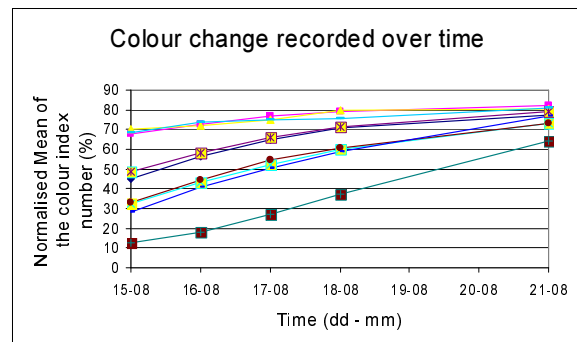


Figure 5: The development of the average colour of nine tomatoes as they change over time as recorded by the developed measuring device.

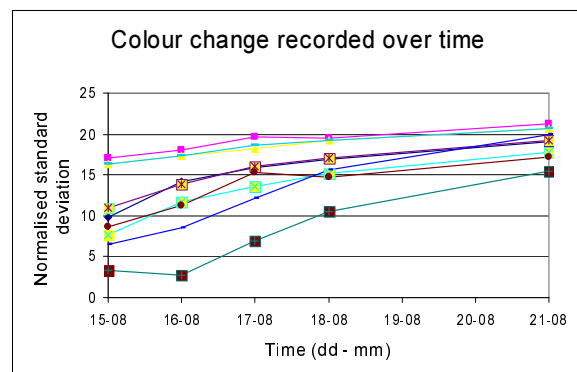


Figure 6: The development of the standard deviation of nine tomatoes as the colour changes over time.

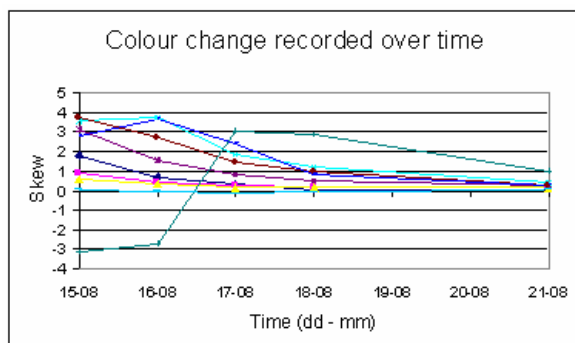


Figure 7: The development of the skew of nine tomatoes as their colour changes over time.

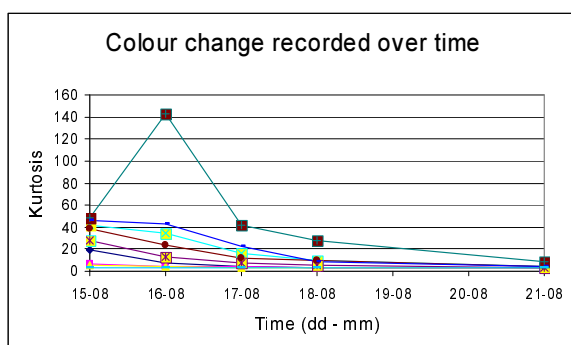


Figure 8: The development of the kurtosis of nine tomatoes as their colour change over time.

In some situations the skew and kurtosis are more difficult to interpret. Since they are 3rd and 4th order moments, they tend to be very sensitive to small fluctuations. This might obscure the outcome in certain cases. In general the outcome is good, see Figures 7 and 8. However in some cases an outlying kurtosis number is found, as for the tomato marked with a '+' in Figure 8 on 16-08 for instance.

With the limes the outcome was less good. Since the limes are stored under normal conditions, without cooling, they began dehydrating. This caused the limes to maintain their green colour, while the rind turned hard. The results out of this test were therefore less useful, other than to obtain more experience with the measuring device.

5 Conclusions

Comparing the outcome with the use of control charts, there is certainly an improvement. The number of classes into which fruit may be graded was increased significantly, while maintaining a correct classification.

The use of the four index numbers, mean, standard deviation, skew and kurtosis, gave a clear indication about the colour distribution of the object. It also allows an easy comparison using the automatic generated graphs between different objects.

The relatively slow measurement speed precludes the use of this prototype from commercial application.

However, the focus of this project is more on providing a useful research tool, and the slow measurement speed is not as important.

6 Further Research

During this research it is demonstrated that the method described is useable. Therefore this method will be applied in further research in which two aspects will be investigated.

- The recording of the colour change needs to be compared to measurements made using a spectrometer.
- A larger study with more objects stored under different temperature regimes to develop kinetic models that can be used to describe colour change a range of constant or varying storage conditions.

The results of this research will be presented at a later stage.

7 Acknowledgements

I would like to acknowledge Massey University for giving me the possibility to learn much during my internship.

8 References

- [1] Y. Edan, H. Pasternak, I. Shmulevich, D. Rachmani, D. Guedalia, S. Grinberg, and E. Fallik, "Color and firmness classification of fresh market tomatoes", *Journal of Food Sciences*, **62**:(4) 793-796 (1997).
- [2] G. Jahns, H.M. Nielsen, and W. Paul, "Measuring image analysis attributes and modelling fuzzy consumer aspects for tomato quality grading", *Computers and Electronics in Agriculture*, **31**:(1) 17-29 (2001).
- [3] E. Bosquez-Molina, J. Domínguez-Soberanes, L.J. Pérez-Flores, F. Diáx-de-León-Sánchez, and J. Vernon-Carter, "Effect of edible coatings on storage life of Mexican limes (citrus aurantifolia Swingle) Harvested in two different periods", in *XXVI IHC - Citrus, Subtropical and Tropical Fruit Crops*, **632**: 329-335 (2004).
- [4] A. Batu, "Determination of acceptable firmness and colour values of tomatoes", *Journal of Food Engineering*, **61**: 471-475 (2004).
- [5] R. Arias, T. Lee, L. Logendra, and H. Janes, "Correlation of lycopene measured by HPLC with the L, a, b color readings of a hydroponic tomato and the relationship of maturity with color and lycopene content", *Journal of Agricultural and Food Chemistry*, **48**: 1697-1702 (2000).
- [6] S. Brandt, Z. Pék, É. Barna, A. Lugasi, and L. Helyes, "Lycopene content and colour of

- ripening tomatoes as affected by environmental conditions", *Journal of the Science of Food and Agriculture*, **86**: 568-572 (2006).
- [7] A. Wold, H.J. Rosenfeld, K. Holte, H. Baugerød, R. Blomhoff, and K. Haffner, "Colour of post-harvest ripened and vine ripened tomatoes (*Lycopersicon esculentum* Mill.) as related to total antioxidant capacity and chemical composition", *International Journal of Food Science and Technology*, **39**: 295-302 (2004).
- [8] S.K. Clinton, T.M.W. Boileau, and J.W. Erdman, "Effect lycopene or tomato powder upon prostate cancer (correspondence)", *Journal of the National Cancer Institute*, **96**:(7) 554-555 (2004).
- [9] P.H. Gann and F. Khachik, "Tomatoes or Lycopene versus prostate cancer; Is evolution Anti reductionist (Editorial)", *Journal of the National Cancer Institute*, **95**:(21) 1563-1565 (2003).
- [10] N.I. Krinsky and E.J. Johnson, "Carotenoid actions and their relation to health and disease", *Molecular Aspects of Medicine*, **26**: 459-516 (2005).
- [11] T. Pranamornkith, J.A. Heyes, and A.J. Mawson, "Effects of CA and alternative postharvest treatments of lime (*Citrus latifolia* Tanaka) fruit", in *International Controlled Atmosphere Research Conference*, Michigan State University, USA, (July 5-10, 2005).
- [12] United States Department of Agriculture, *United States standards for grades of fresh tomatoes*. 1997.
- [13] M.M. Lana, *Modelling quality of fresh-cut tomato based on stage of maturity and storage conditions*. Horticultural Production Chains Group, Plant Science. Ph.D. Thesis. Wageningen: Wageningen University. (2005).
- [14] A. Lopez Camelo and P.A. Gomez, "Comparison of color indices for tomato ripening", *Horticultura Brasileira*, **22**:(3) 534-537 (2004).
- [15] R. Gómez, J. Costa, M. Amo, A. Alvarruiz, M. Picazo, and J. Pardo, "Physicochemical and colorimetric evaluation of local varieties of tomato grown in SE Spain", *Journal of the Science of Food and Agriculture*, **81**: 1101-1105 (2001).
- [16] G. Polder, G.W.A.M. van der Heijden, and I.T. Young, "Hyperspectral image analysis for measuring ripeness of tomatoes", in *2000 ASAE International Meeting*, Milwaukee, Wisconsin, (July 9-12, 2000).
- [17] G. Polder, G.W.A.M. van der Heijden, H. van der Voet, and I.T. Young, "Measuring surface distribution of carotenes and chlorophyll in ripening tomatoes using image spectrometry", *Postharvest Biology and Technology*, **34**: 117-129 (2003).
- [18] G. Polder, G.W.A.M. van der Heijden, and I.T. Young, "Tomato sorting using independent component analysis on spectral images", *Real-Time Imaging*, **9**: 253-259 (2003).
- [19] D.G. Bailey and R.M. Hodgson, "VIPS - a Digital Image Processing Algorithm Development Environment", *Image and Vision Computing*, **6**: 176-184 (1988).
- [20] M.L.A.T.M. Hertog, J. Lammertyn, M. Desmet, N. Scheerlinck, and B.M. Nicolai, "The impact of biological variation on postharvest behaviour of tomato fruit", *Postharvest Biology and Technology*, **34**: 271-284 (2004).
- [21] F. Artes, E. Sanchez, and L.M.M. Tijskens, "Quality and shelf life of tomatoes improved by intermittent warming", *Lebensmittel-Wissenschaft und-Technologie*, **31**: 427-431 (1998).

Athlete Performance Video Overlay

S. Sarjeant¹, R. Green¹

¹University of Canterbury, Department of Computer Science and Software Engineering

Email: srs67@student.canterbury.ac.nz

Email: richard.green@canterbury.ac.nz

Abstract

This paper sets out to detail novel algorithms related to calculating and displaying robust athlete performance data overlaid on video. The paper describes a robust background segmentation algorithm that enables human body performance parameters to be calculated. A further algorithm is presented that can successfully recover from stereo camera deficiencies. Athlete performance parameters include center of mass, principal axis, speed, acceleration, cyclic motion and energy usage. The motivation for calculating human performance parameters is to aid movement disorder clinicians, coaches and athletes.

Keywords: computer vision, machine vision, motion analysis, human body, sport coaching

1 Introduction

The goal of this paper is to present algorithms that are relevant when determining motion characteristics of a human moving naturally within a scene. The justification is that determining such characteristics allows sports coaches and movement disorder clinicians to gain a more accurate vision of how a human is moving and where to focus attention. The paper will first detail prior research and the process involved in estimating body characteristics. It will then detail the algorithms researched, finishing with an evaluation of this research. Several motion parameters and their algorithms are presented including center of mass, principal axis, axial aligned bounding boxes, speed, acceleration and cyclic motion.

2 Background

2.1 Background Segmentation

Background segmentation is the process of segmenting the background from the desired foreground objects within the scene, in this case a human. Temporal differencing algorithms are widely in use due to their adaptiveness with dynamic backgrounds [1], two examples are: Adjacent Frame Difference Algorithm (AFDA) [2] and Double Difference Algorithm (DDA) [3], which is used in this research.

2.2 Contact & Non-Contact

In detecting key body joints systems thus far fall into two categories, contact and non-contact. Contact systems are when the user has various joint

marker sensors attached to their body, compared to non-contact in which the user is without any sensors attached and can act freely and more naturally using computer vision based motion tracking. Due to the expense and cumbersome use of contact systems this paper will focus on non-contact computer vision based methods.

2.3 Human Model

Determining a human model from computer vision is a widely researched area with [4] providing an overview of the field. Two areas of interest exist, model reconstruction and movement recognition. Constructing a human model can be done through various methods such as blob segmentations [5] and distance transformations [6]. Whilst movement recognition can be completed through the use of Hidden Markov Models (HMM), [7] for example, which are used to recognise sign language.

2.4 Tracking

Motion tracking involves keeping track of coordinates of interest on the subjects body. Mathematical models exist for tracking and predicting coordinate locations in successive frames such as the Kalman filter [8], which uses state estimation based on the Gaussian distribution, and the Condensation filter [9] or Particle Filter [10], which uses conditional density propagation with the posterior distribution.

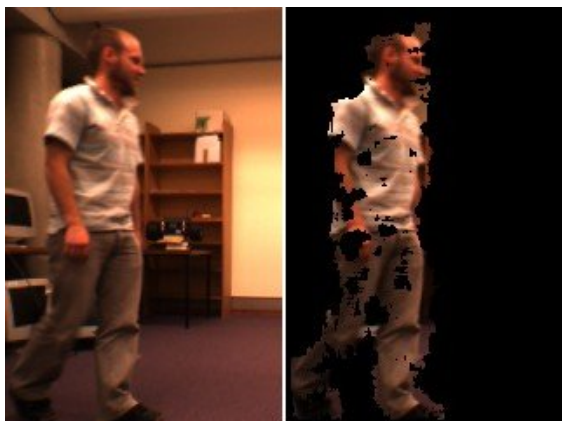


Figure 1: Example of background segmentation using disparity filtering.

3 Method

3.1 Background Segmentation

For this project the Double Difference Algorithm (DDA) served as the basis for an adaptive disparity segmentation algorithm. The process first involves performing the DDA algorithm on the incoming video frame and then computing the center of mass of the segmented region. The disparity of the pixel located at the center of the mass is then used to filter the disparity data provided by the stereo camera. The subsequent body parameter algorithms then operate on this filtered disparity image.

3.2 3D Data

To cope with the Bumblebee's inability to reliably determine the disparity of an individual pixel, a simple yet robust algorithm was implemented. Instead of checking for the disparity of the exact required location, a region of interest (ROI) is constructed and surveyed around that pixel. This method surveys disparity pixels within the constructed square and then returns the average of the detected values.

3.3 Athlete Overlays

All athlete overlay calculations operate on the binarized segmented disparity image.

3.3.1 Center of Mass

The center of mass is the most robust body parameter available and is used in segmenting the moving object and to estimate other body parameters including principal axis, speed and cyclic motion. The center of mass (\bar{x}, \bar{y}) [11], is considered as:

$$\bar{x} \sum_{i=1}^n \sum_{j=1}^m B[i, j] = \sum_{i=1}^n \sum_{j=1}^m j B[i, j] \quad (1)$$

$$\bar{y} \sum_{i=1}^n \sum_{j=1}^m B[i, j] = \sum_{i=1}^n \sum_{j=1}^m i B[i, j] \quad (2)$$

3.3.2 Principal Axis

To correctly determine the principal axis of a shape it must be elongated, and the principal axis is thus considered the axis of least inertia. From [11], [12], the principal axis is described as:

$$\tan^2 \theta + \frac{\mu_{20} - \mu_{02}}{\mu_{11}} \tan \theta - 1 = 0 \quad (3)$$

Where the second order moments μ_{20} , μ_{11} and μ_{02} are considered as:

$$\mu_{20} = \sum_{i=1}^n \sum_{j=1}^m B[i, j] (x_{ij} - \bar{x})^2 \quad (4)$$

$$\mu_{11} = \sum_{i=1}^n \sum_{j=1}^m B[i, j] (x_{ij} - \bar{x})(y_{ij} - \bar{y}) \quad (5)$$

$$\mu_{02} = \sum_{i=1}^n \sum_{j=1}^m B[i, j] (y_{ij} - \bar{y})^2 \quad (6)$$

Solving equation 3 yields:

$$\theta = \frac{1}{2} \arctan \left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right) \quad (7)$$

3.3.3 Bounding Boxes

The system constructs and displays a two dimensional bounding box around the detected area of movement. To produce the most accurate bounding box it is aligned with the principal axis and thus the accuracy of the bounding box depends on the accuracy of the principal axis. Generating an axis aligned bounding box involves calculating the maximum widths of the human at angles orthogonal to the principal axis. It also generates axis aligned bounding boxes for the upper and lower segments of the human as determined by the center of mass.

3.3.4 Cyclic Motion

To determine the cyclic motion of a human as they walk in front of the camera the visible area of their lower body is calculated. The lower body is considered as anything below the center of mass. Other methods to determine cyclic motion were also tested; the distance between each feet, and the principal axis of the lower body, but neither

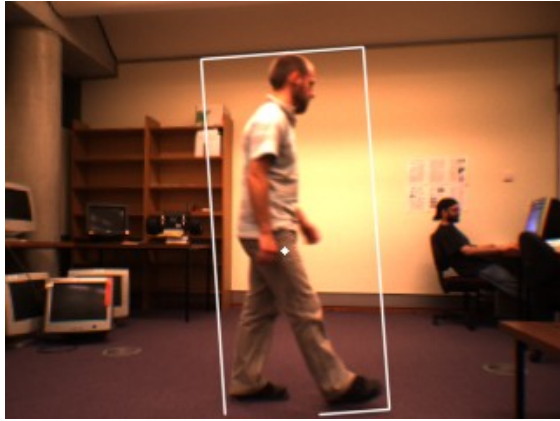


Figure 2: A principal axis aligned bounding box, with center of mass also shown.

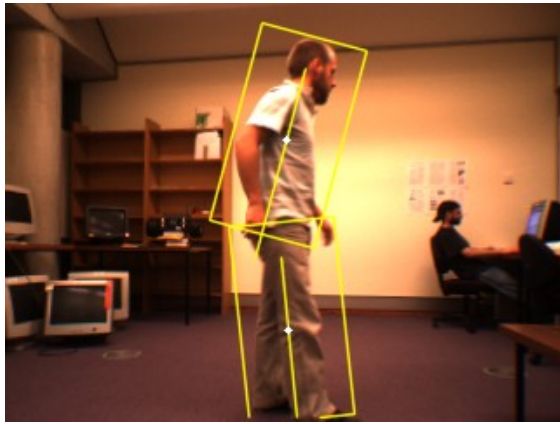


Figure 3: The principal axis and bounding boxes are shown for the upper and lower body.

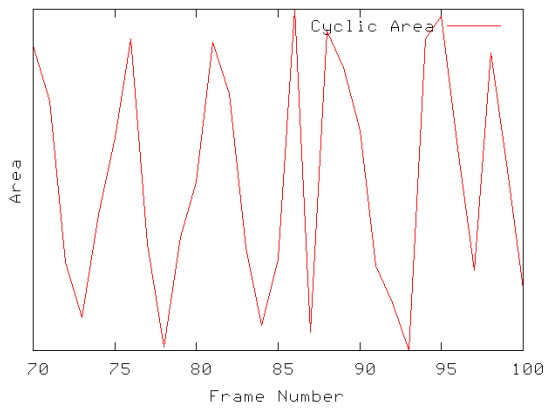


Figure 4: Graph of cyclic motion of gait moving perpendicular to the camera.

performed sufficiently accurate. Mathematically, the cyclic motion, c , is represented as:

$$c = \sum_{i=\bar{x}}^n \sum_{j=\bar{y}}^m B[i, j] \quad (8)$$

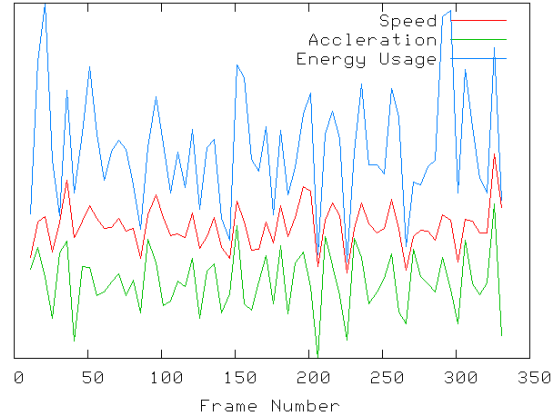


Figure 5: Graph of the speed (red), acceleration (green) and energy usage (white) of a human as they move within a scene. Y axis values have been compressed to fit on the same graph.

3.3.5 Speed & Acceleration

With 3D data and the center of mass located calculating the speed, s , and acceleration, a , of the moving object is a trivial task. The Euclidean distance between the objects center of mass position is calculated from the current frame i and the previous frame $i - 5$. This distance then enables the speed of movement to be calculated as well as acceleration. In this implementation both the speed and acceleration are updated every 5 frames of input images.

$$x_d = (x_i - x_{i-5})^2 \quad (9)$$

$$y_d = (y_i - y_{i-5})^2 \quad (10)$$

$$z_d = (z_i - z_{i-5})^2 \quad (11)$$

$$s = \frac{\sqrt{x_d + y_d + z_d}}{5} \quad (12)$$

Acceleration is then considered as:

$$a = s_i - s_{i-5} \quad (13)$$

3.3.6 Energy Usage

A simplified model of the energy used, e , by the moving human is calculated and graphed. The current area of the human is multiplied by the speed at which the human is moving to produce this energy usage result.

$$e = s \times \sum_{i=1}^n \sum_{j=1}^m B[i, j] \quad (14)$$



Figure 6: Example of disparity filtering (left) and DDA (right).

3.4 Setup

The system was implemented in C++ on an Intel Pentium IV, 2.4GHz PC with 512MB of memory. The Bumblebee stereo camera developed by Point Grey Research was utilised. The Intel Open Computer Vision Library and the Triclops/Digiclops Library for stereo processing provided by Point Grey Research aided implementation. Video was retrieved at a resolution of 320x240.

4 Results

4.1 Performance

The setup was described in section 3.4 and would perform in real-time. If the user required the system to calculate all parameters at a given time performance would degrade below real-time.

4.2 Background Segmentation

A formal evaluation of our background segmentation algorithm has not been completed for this paper, however it did appear to perform better than DDA by itself and an example frame comparison is shown in figure 6. An evident advantage disparity filtering has over sole DDA is that slow moving or stationary objects do not fade into the background.

4.3 3D Data

Due to the heavy reliance the performance algorithms place on accurate 3D data this has been evaluated. The center of mass is used as the pixel of interest in figures 7, 8 and 9.

Figures 7, 8 and 9 show that the square average approach aided in estimating the real world coordinates of pixels the camera was unable to determine. Of the 335 frames captured the average approach successfully recovered from the 12% of frames that

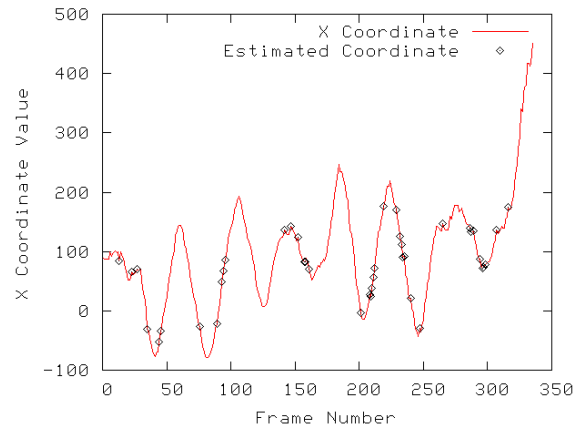


Figure 7: X coordinate of a moving object. Black circles indicate an estimated value.

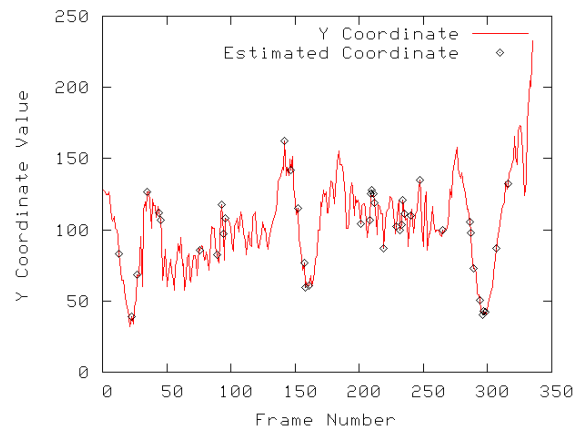


Figure 8: Y coordinates of a moving object. Black circles indicate an estimated value.

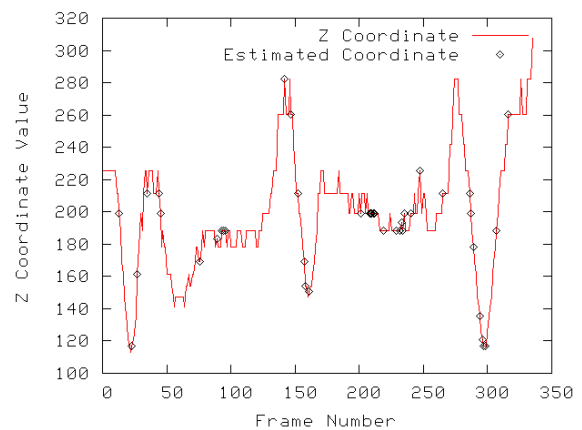


Figure 9: Z coordinates of a moving object. Black circles indicate an estimated value.

the exact world coordinates of the point could not be determined. The figures illustrate that the estimated values generally fit on the curve to the next detected coordinate. Further improvements in stereo camera hardware and stereo processing algorithms will aid in negating this issue.

4.4 Conclusions & Future Work

This paper has presented algorithms relating to segmenting and calculating athlete performance data. The system was designed using non-contact computer vision based techniques without the need for explicit initialisation by the user. A novel background segmentation algorithm was researched that works robustly in cluttered environments, for which a formal evaluation could be completed for future research.

An accurate algorithm to estimate the 3D coordinate of a point when the camera is unable to was also detailed. This algorithm successfully recovered from the 12% of frames when the camera failed to determine the 3D coordinate of a selected point.

Currently the system can only handle one moving object within the scene, it would be beneficial to research the use of clustering algorithms to further segment regions of movement. This would allow subsequent algorithms to process and calculate data on all moving objects.

The current method for calculating the cyclic motion of a human relies on the human moving along the x axis in front of the camera and will fail when the human walks back and fourth along the z axis. Further research needs to be conducted to find a simple algorithm that works correctly in both cases.

Motion tracking algorithms such as those described in 2.4 could be researched and implemented to further improve the robustness of the system.

References

- [1] M. Piccardi, "Background subtraction techniques: a review," in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, vol. 4, 2004.
- [2] A. J. Lipton, H. Fujiyoshi, and R. S. Patil, "Moving target classification and tracking from real-time video," in *Applications of Computer Vision, 1998. WACV '98. Proceedings., Fourth IEEE Workshop on*, pp. 8–14, 1998.
- [3] J. K. Aggarwal and Q. Cai, "Human Motion Analysis: A Review," *Computer Vision and Image Understanding: CVIU*, vol. 73, no. 3, pp. 428–440, 1999.
- [4] T. B. Moselund and E. Granum, "A Survey of Computer Vision-Based Human Motion Capture," *Computer Vision and Image Understanding: CVIU*, vol. 81, no. 3, pp. 231–268, 2001.
- [5] C. R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-Time Tracking of the Human Body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780–785, 1997.
- [6] S. Iwassawa, J. Ohya, K. Takahashi, T. Sakaguchi, K. Ebihara, and S. Morishima, "Real-Time Estimation of Human Body Posture from Trinocular Images," in *International Workshop on Modeling People at ICCV'99*, 1999.
- [7] T. Starner and A. Pentland, "Real-Time American Sign Language Recognition From Video Using Hidden Markov Models," in *SCV95*, p. 265, 1995.
- [8] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME-Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.
- [9] M. Isard and A. Blake, "CONDENSATION – conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 2, pp. 5–28, 1998.
- [10] J. Deutscher, A. Blake, and I. Reid, "Articulated body motion capture by annealed particle filtering," in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, vol. 2, pp. 126–133, 2000.
- [11] R. Jain, R. Kasturi, and B. G. Schunk, *Machine Vision*. United States of America: McGraw-Hill, 1995.
- [12] K. Lee and R. D. Green, "Temporally Synchronising Image Sequences using Motion Kinematics.," in *Proceedings of Image and Vision Computing New Zealand 2005*, 2005.
- [13] Q. Zang and R. Klette, "Robust background subtraction and maintenance," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 2, pp. 90–93, 2004.
- [14] D. M. Gavrila, "The Visual Analysis of Human Movement: A survey," *Computer Vision and Image Understanding: CVIU*, vol. 73, no. 1, pp. 82–98, 1999.

Image Processing of Meat Images for Visible/Near Infrared Spectroscopy Reference

Lee Streeter^{1,2}, G Robert Burling-Claridge², Michael J Cree¹

¹University of Waikato, Dept. Engineering, Hamilton, New Zealand.

²AgResearch, Ruakura Campus, Hamilton New Zealand.

Email: lvs2@phys.waikato.ac.nz

Abstract

The segmentation of meat images for reference in spectral analysis is considered in this paper. The purpose is to compute the proportions of the major visible constituents of lean fat and connective tissue. The segmentation of the meat images primarily utilised colour information, for which the RGB colour space and an adaptation of a YUV like colour space were primarily used. The ADTree algorithm was used to optimise specificity of fat object detection. Localisation of spectra acquisition positions in the meat images was done using other specially acquired images. Canonical correlations were used to perform preliminary comparison of spectra to the image processing result. Strong correlations were observed between spectra and visible lean and fat. Weaker but significant correlations were observed for the visible connective tissue.

Keywords: Meat Image Segmentation, Near Infrared Reflectance Spectroscopy, Reference Method

1 Introduction

In this paper we investigate image processing of meat to obtain reference data for Visible/Near Infrared Reflectance (Vis/NIR) spectroscopy. The objective was to segment meat images into background, lean, fat and connective tissue. The segmentation result was compared to corresponding Vis/NIR spectra. This work is a continuation of the work presented in [1], in which preliminary processing measures were considered. Processing of meat images is not new in the literature [2]. Processing of meat images for spectroscopic reference appears to be.

The Meat Quality project at AgResearch is a FRST funded initiative with the ongoing goal of finding better ways to objectively assess meat quality. Vis/NIR is being employed as a possible commercial tool for non-destructively measuring the chemical and physical properties of meat. Vis/NIR spectroscopy has traditionally only scanned a single point or small region at a time. Spectroscopic imaging (usually referred to as hyperspectral imaging in the literature) provides spectral information at a number of locations, yielding extra information. Hyperspectral imaging systems are being developed, but as yet commercially available systems either have limited spectroscopic bandwidth or considerably limited spectral resolution (these cases are usually referred to as multispectral imaging). Given a

suitable hyperspectral imaging device, there is considerable room for research into the fusion of traditional NIR spectroscopic data analysis and image processing techniques.

As part of the Meat Quality research, AgResearch is looking at localised properties throughout the volume of *m. longissimus dorsi* (porterhouse, rumpsteak and ribeye). This “3-D mapping” is hoped to reveal new information about spatial variability in the characteristics of the muscles. Image processing is being investigated to establish localised ‘truth’ of meat content (fat-lean ratios, etc) for the Vis/NIR spectral analysis. This paper describes the image processing routine developed.

The paper is structured as follows. Section 2 outlines the data acquisition process. Section 3 details the image processing methodology employed to segment the images. Section 4 describes the processing of special images to link the image processing with the spectra and statistics used to compare the spectra with image processing. Section 5 gives results and section 6 conclusions.

2 Data Acquisition

For the work outlined in this paper the data acquisition process is of great importance. The images under consideration are of meat, harvested according to a specific protocol, established in consultation with meat science expertise. Measures were taken to regulate the data acquisition process.

Four pairs of beef *m. longissimus dorsi* (LD) were harvested from steers shot with captive bolt. No electrical stimulation was used to speed induction of rigor mortis. The LD muscles were wrapped in cling-foil to prevent shortening [3] and placed in a holder tube. The wrapped LD muscles were stored at 15 degrees Celsius until 24 hours post rigor mortis (approximately 72 hours). At 24 hours post rigor the muscles (now meat) were transferred to a custom built holding and positioning tube for data acquisition. This holding and positioning tube allowed for the acquisition of images and spectra with constancy and regularity.

The images were acquired through a JAI machine vision colour camera interfaced through a Matrox capture card. The image size was 760×570 pixels. The host computer was running Windows XP with legacy image capture software for the Matrox card. A 14mm slice was taken and discarded, leaving a new meat face for data acquisition. After data were acquired, the slicing process was iterated until twenty meat faces were examined. At least three images were captured of each meat slice, the best image later selected for processing. Concurrent with imaging of the meat face, the white 90% reflectance and grey 18% reflectance sides of a Jessops grey card were also imaged. These grey card images had two uses. The first use was before image acquisition the grey 18% was used to check and calibrate the colour balance of the camera. The second use was shade correction in image pre-processing (section 3).



Figure 1: A meat slice image.

After image acquisition forty nine spectra were taken over the face of the meat slice in a 7×7 grid. The spectra were taken with a KES spectrometer with an optical fibre probe. This probe was held in front of the meat face by an appendage of the meat holder rig. This appendage allowed spectra to be taken with 14mm horizontal and vertical spacing. Spectra taken by this procedure formed spatially coarse hyperspectral images. A blue disc

of cardboard was placed in the probe holder and images were taken at each relevant position. The blue disc images were used to match each NIR spectrum with the corresponding image location.

3 Image Processing Methodology

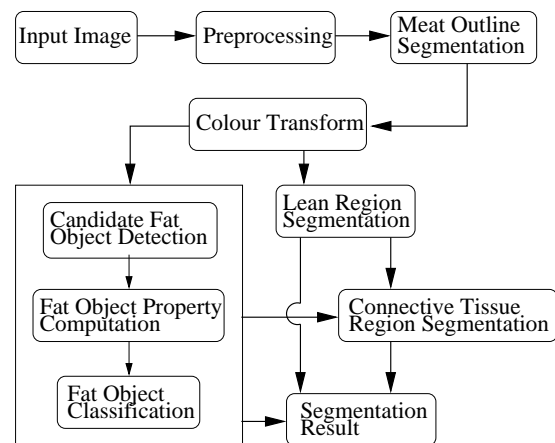


Figure 2: Flow Diagram of the Meat Segmentation Image Processing.

The processing of the meat images is designed to segment the meat images into major constituent parts (background, lean, fat and connective tissue). Visual examination of the images made the task appear deceptively simple. Some confounding factors were present in the images. There were issues due to low illumination levels at image acquisition and fluid which produced objects similar in appearance to fat marbling. The idea behind the processing was to use many small steps to gradually segment the image, resulting in a robust procedure. Figure 2 shows a flow diagram of the processing scheme. Figure 1 shows an example image of a meat slice. Note the metal ring which is the end of the tube used to hold the meat, the presence of intra and extra muscular fat and specularities on the dark red lean due to moisture.

3.1 Preprocessing

The images of the grey 18% Jessops were used as reference for background shading. First an 11×11 median filter was used on each colour plane in the grey images to remove any scratches or marks that appeared time. Then the meat images were divided pixel-wise by the grey image.

Reflections due to liquid secretion appear as small roundish white objects and resemble fat. They represented a confounding factor to segmentation of the meat images. To reduce this effect the meat face was padded down lightly with a tissue to absorb the fluid. Despite this padding, some specu-

larity is still apparent. A 5×5 pixel median filter was used to reduce the specularity.

The object of interest, i.e. the meat face in the meat holder, was positioned in the same place in each image. A circle aperture mask was used to mask out everything up to and including the meat holder tube. Before processing the outline of the meat object boundary was estimated in a two step process. The first step removed most of the background not already masked out but was slightly underspecific. The second step was used to ‘chop away’ what was left by the first. The first step proceeded as follows:

- The difference between the green and red planes was computed. A threshold of greater than or equal to zero was applied to the difference image.
- All objects that did not reach the boundary were removed. The result was inverted providing candidate objects for the meat.
- The largest candidate object was selected and each colour plane masked (see figure 3, left).

The second step proceeded as follows:

- A threshold of 0.2 was applied to the masked red plane.
- To fill holes, a floodfill operation was applied to the resultant binary image.
- The largest binary object was selected as the meat object mask. Each colour plane was subsequently masked (see figure 3, right).



Figure 3: Preliminary binary meat object mask (left) and final meat object mask (right).

3.2 Processing

The processing scheme is designed to segment the meat into lean, fat and connective tissue in that order. For colour segmentation an adaptive linear colour transformation was used (a modified version of that presented in [4]). The colour transformation first found the first eigenvector \underline{v}_1 of the pixel colour in the meat region. A transform matrix \mathbf{T} was formed as

$$\mathbf{T} = \begin{bmatrix} 1 & 1 \\ \underline{v}_1^t & -2 & 1 \\ 1 & 1 \end{bmatrix} \quad (1)$$

which was subsequently orthogonalised by the Gram-Schmidt process [5]. The orthogonalised version of \mathbf{T} provided an adaptive transformation from RGB to a YUV like colour space.

$$\mathbf{Y'U'V'} = \mathbf{T} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2)$$

Lean region segmentation was done in the U' plane. The purpose of the lean region segmentation was to identify the boundary of the lean object. Differentiation between the lean and extramuscular fat was desired. Differentiation between lean and intramuscular fat was part of the fat segmentation, which occurred after the lean segmentation. The procedure was

- The U' plane was thresholded at threshold value t found as

$$t = \bar{U}' + x\sigma_{U'} \quad (3)$$

where \bar{U}' and $\sigma_{U'}$ are the U' plane mean and standard deviation of all pixels within the meat holder. The multiplier x was typically set to 0.15, but had to be adjusted for some images.

- The binary result of thresholding was tidied up by a morphological closure with a disc structuring element of radius one (a diamond shape) and a flood fill to fill holes.
- Some extraneous objects were removed by masking out the background according to the background mask computed previously.
- The largest binary object was selected as the lean object.

The red plane of Figure 6 shows the result of lean object segmentation. Note the ‘arm’ off the right hand side which is due to blood seepage into the cling wrap around the meat.

Fat object detection consisted of detection of candidate fat objects, computation of some shape and colour features on the candidate objects and classification of fat objects. Candidate fat object detection was very similar to the lean object detection but with the following differences: processing was carried out in the Y' plane; in computing the threshold, the standard deviation was multiplied by 0.4 (cf. equation 3) and instead of closing and



Figure 4: Candidate fat object binary image.

floodfilling, an opening was used with a disc structuring element of radius two pixels.

The candidate fat object detection typically yielded a large number of false candidates (see figure 4, cf. figure 1). To improve specificity of fat object detection a set of features were computed and pattern classification was employed. The features computed included: the area in pixels of each object; the mean and standard deviation of each plane in: the RGB images; CIE La*b* colour space and HSV colour space. In total nineteen features were computed.

The ADTree algorithm [6] was used to classify candidate fat objects as fat or spurious objects. A tree based method was selected because there are two types of fat object present: large extramuscular fat and smaller intramuscular fat blobs. The ‘branching’ of a tree algorithm allows the handling of multiple types of objects that belong to the same broad class. ADTree is very flexible, allowing multiple branches per node. Also ADTree returns a numerical value for thresholding rather than a hard class result.

To train the ADTree, the candidate detection algorithm was run on every fourth image in the entire image set (thirty of the one hundred and twenty images). For the candidate objects the true classification was set manually and the feature data computed. The ADTree algorithm was trained in WEKA [7], a data mining package written in Java. Ten fold cross validation was used. Classification of candidate fat objects gives rise to the question of how many false detections are present in each image. Free Receiver Operator Characteristics [8] (FROC) provides quantitative assessment of the false detection per image rate. Thus FROC metrics were used to assess the quality of the training and testing. The reader is reminded that the goal here was *not* to develop an algorithm for use on subsequent data sets, rather the goal was to provide suitable reference values for analysis of corresponding NIR spectra. Thus the most important test was to compare the image processing result with the NIR spectra. The green plane of figure 6 shows the result image of candidate fat object classification.

Connective tissue is present in all muscular structure. It is found between muscles and between muscle and fat. In the images it appeared dark and glassy. The lean and fat images were used to mask out all pixels clearly not connective tissue. Such masking simplified detection of connective tissue. The procedure was as follows:

- Binary logical OR image $o = \text{lean OR fat}$, o identifies all pixels already assigned a class.
- The lean and fat images were dilated with a disc structuring element of radius two, on which a logical AND image a was computed. This identified all lean-fat boundaries where connective tissue must be.
- A third image, c , formed by filling the holes in o and heavily closing with a disc structuring element of radius 10.
- An initial possible connective tissue binary image was formed as $t_0 = (\text{NOT}(o) \text{ OR } a) \text{ AND } c$ (see figure 5 left).
- A more precise candidate connective tissue image was found as $t_1 = (R \geq 0.6) \text{ AND } t_0$ where R is the red plane of the meat image.
- The candidate connective tissue images had significant ‘cut-ins’ into the lean region. These were removed by first closing the binary lean region image with a disc structuring element of radius 9, then eroding it with a disc of radius 5. The resultant morphologically transformed image l_m was used to mask the candidate connective tissue image $t_{\text{final}} = t_1 \text{ AND NOT}(l_m)$ (see figure 5, right).



Figure 5: Connective tissue computation.

The result of processing at this point was three binary images classifying regions into the three image constituents of interest. However there was significant overlap between regions. An hierarchical class preference was used to assign final classification. Connective tissue was the hardest to detect and typically was impinged upon by other classes. Thus Connective tissue was assigned highest priority. The intramuscular fat was by definition embedded in the lean. Thus it was necessary to make fat second in priority and lean last. Figure 6 shows the final result (cf. figure 1).

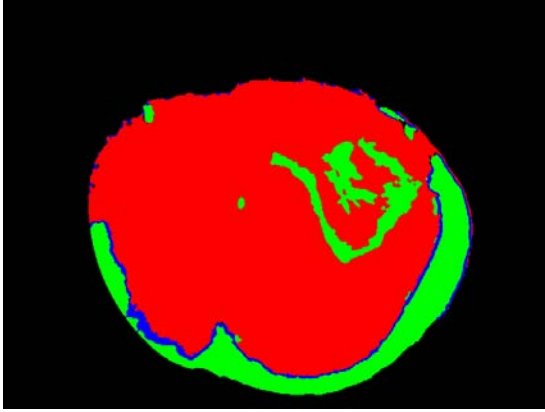


Figure 6: Final segmentation result image.

4 Linking and Comparing the Images with the Spectra

4.1 Processing the Blue Spot Images

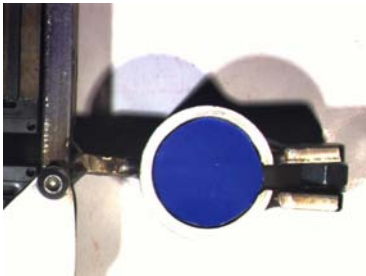


Figure 7: A blue spot image. Here the probe holder is visible with a blue disc in place of the spectrometer probe. White paper has been placed between the probe holder and the meat face to provide a suitable image background.

Forty nine blue spot images were taken, each with the blue spot at one spectrum acquisition position. Figure 7 shows an example blue spot image. Detection of the blue spot as the largest bright blue object was trivial. By detecting the position of the blue spot in each image the result of the meat image processing was localised and matched with each spectrum.

In some of the images the blue spot was partially excluded at the image boundary. This made the rigorous determination of the centre of the blue spot non-trivial. A rigorous method to find the blue spot centre was devised. The blue spot was detected by thresholding. The binary blue spot images were perimeterised to yield the visible boundary. A model of the circular blue spot object perimeter was matched to the blue spot location using a simple scheme based on the Hausdorff distance [9]. The Hausdorff distance is a measure of distance between two sets of vertexes (pixel locations in this case). The model was iteratively shifted around the image. The location of the blue

spot centre was determined by the position of the model that minimised the Hausdorff distance.

4.2 Comparing of Image Processing with the NIR Spectra: Canonical Correlations

Canonical correlations [10] was used to compare the Vis/NIR spectra with the image processing. Here we had NIR spectra S and image processing reference values I for lean, fat and connective tissue. Canonical correlations finds the bases w_S and w_I such that

$$\rho = \frac{E[w_S^T S^T I w_I]}{\sqrt{E[w_S^T S^T S w_S] E[w_I^T I^T I w_I]}} \quad (4)$$

is maximised. Here $E[\cdot]$ is the expected value and $[\cdot]^T$ is the matrix transpose operator. The number of values for ρ is determined by the minimum number of values in the data examined. Since there were three reference values in I (lean etc) and one hundred and twenty two wavelengths in the spectra, we had three values for ρ . Canonical correlations were computed for each reference value in turn. Correlations could have been found for all three references at once but inspection of the spectra showed significant swamping of the effect due to connective tissue. First canonical correlations were taken over all data instances (direct). Then the data were divided up per animal into three sets and the basis vectors w_i were computed on two animals and applied to the third in sequence (cross validated). Cross validation of canonical correlations is analogous to linear regression. Significance testing of all correlations was done.

5 Results and Discussion

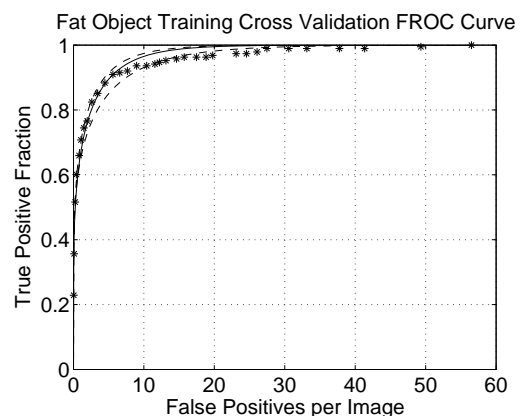


Figure 8: FROC curve for ADTree training cross validation of candidate fat objects. A fitted theoretical FROC curve with 95% confidence intervals is shown.

	Lean	Fat	Connective Tissue
Direct	0.97	0.96	0.84
Cross Validated	-0.86	0.88	0.60

Table 1: Correlations of the image processing result with the spectra.

Figure 8 shows the FROC curve for training and testing the fat detection classification. A threshold value of 0.2154 was chosen which corresponded to true positive fraction of 0.91 with 5.33 false positives per image. Visual examination of the classified fat images showed in general very few false positives. Some fat objects had a tendency to spill into the lean region. Particularly the extramuscular fat.

Table 1 lists the canonical correlations. The direct correlations were all high and were significant at 99% confidence. The cross validated correlations are lower but also significant at 99% confidence. The cross validated correlations account for variation between animals. Thus they are more realistic than the direct correlations. Regardless the correlations for lean and fat are good. Investigation into Vis/NIR spectral analysis for lean and fat prediction is warranted. The lower correlation for connective tissue is unsurprising given that the contribution to spectra is small. Greater spatial resolution in spectral scanning is necessary to sufficiently assess calibration against connective tissue.

6 Conclusion and Future Directions

The segmentation of meat images for reference in spectral analysis has been outlined. Correlations between the image processing result and corresponding spectra were found. These correlations indicated strong relationships between visible lean and fat with the spectra. Insufficient spatial resolution in spectra for connective tissue was observed. Further analysis of spectra by sophisticated chemometric techniques is warranted.

Increased resolution in spectral scanning is desired. To this end methods for hyperspectral imaging are under investigation. The intended goal is to acquire hyperspectral images of sufficient resolution and to hybridise image processing and spectral analysis techniques. Just what resolution is sufficient remains an open question.

7 Acknowledgements

The authors acknowledge the expert advice of Carrick Devine and Kevin Taukiri and the fabrication

of equipment by Keith Hill. L. Streeter acknowledges the Tertiary Education Commission in providing an Enterprise Scholarship.

References

- [1] L. Streeter, R. Burling-Claridge, and M. J. Cree, "Colour image processing and texture analysis on images of porterhouse steak meat," in *Image and Vision Computing New Zealand*, (Dunedin, New Zealand), pp. 398–343, November 2005.
- [2] T. Brosnan and D.-W. Sun, "Improving quality inspection of food products by computer vision - a review," *Journal of food engineering*, vol. 61, pp. 3–16, 2004.
- [3] C. Devine, N. Wahlgren, and E. Tornberg, "Effect of rigor temperature on muscle shortening and tenderisation of restrained and unrestrained beef m. longissimus thoracis et lumborum," *Meat Science*, vol. 51, pp. 61–72, 1999.
- [4] G. S. Gupta, D. Bailey, and C. Messon, "A new colour space for efficient and robust segmentation," in *Image and Vision Computing New Zealand*, (Akaroa, New Zealand), pp. 315–320, November 2004.
- [5] H. Anton, *Elementary Linear Algebra*. New York, USA: Wiley, seventh ed., 1994.
- [6] Y. Freund and L. Mason, "The alternating decision tree learning algorithm," in *Proc. 16th International Conf. on Machine Learning*, pp. 124–133, Morgan Kaufmann, San Francisco, CA, 1999.
- [7] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*. San Francisco: Morgan Kaufmann, second ed., 2005.
- [8] D. Chakraborty, "Maximum likelihood analysis of free-response receiver operating characteristic (FROC) data," *Medical Physics*, vol. 16, pp. 561–568, 1989.
- [9] D. P. Huttenlocher and W. J. Rucklidge, "A multi-resolution technique for comparing images using the hausdorff distance," tech. rep., Cornell University, 1992. <http://techreports.library.cornell.edu:8081/Dienst/UI/1.0/Display/cul.cs/TR92-1321>, Date Accessed 1/9/2006.
- [10] M. Borga, "Canonical correlations a tutorial." Online Tutorial, 1999. <http://www.imt.liu.se/~magnus/cca/tutorial/>, date accessed 28/08/2006.

Quality Assessment of Retinal Images

Y. Kwon¹, A. Bainbridge-Smith¹, A.B. Morris²

¹Dept. Electrical & Computer Engineering, University of Canterbury.

²Christchurch School of Medicine, University of Otago.

Email: Andrew.Bainbridge-Smith@canterbury.ac.nz

Abstract

This paper investigates the assessment of image quality of retinal images. It is based on automatic detection and localisation of the optic disc to assess that the field of view is sufficiently large. Image sharpness as a measure of focus is used to assess image clarity. We report on the development of these algorithms and results obtained from a sample test database of 100 images. The intent of the work is to automate the quality assessment of an image database with an excess of 100,000 images. This forms the initial step of a larger research programme for computer assisted screening of diabetic retinopathy images.

Keywords: Image Quality, Medical Imaging, Retina

1 Introduction

Vision is arguably the greatest of the human senses and its loss brings significant costs, both to the individual and to society. The leading causes of blindness in working age people is due to diabetic complications [1]. Control of blood sugar is the most important mechanism to controlling the disease. However, if ocular complications occur (termed diabetic retinopathy) specific ophthalmic treatments exist. Early intervention and constant monitoring is essential, with an estimated 90% of visual loss being avoidable if followed[2].

Monitoring takes the form of regular retinal examinations; digital images of the retina are taken and graded by a trained professional, to assess the level of severity. Severity in turn determines the frequency of examinations, between 2 yearly and 3 months, and also when treatment occurs. However, the specific details of monitoring and the desire for a national New Zealand Screening Program[3] have given rise to a number of problems for which medical imaging and image processing of the retina are desirable. Central to these problems is New Zealand's dispersed population, the large quantity of images being obtained, together with a significant shortage of professionals to read them.

The solution, potentially, is some form of computer assisted screening. This paper describes preparatory work for a research programme to address this issue. The current diabetic retinal image database at Canterbury District Health Board (CDHB) consists of some 100,000 plus images. As complex computer analysis is anticipated in the future research work, one must ensure that the images used

for processing are of sufficient quality. This paper therefore describes work for assessing image quality of full-colour retinal images acquired using a mydriatic digital camera. The objective being to produce a smaller subset of images, in the current database, with sufficient quality for further analysis.

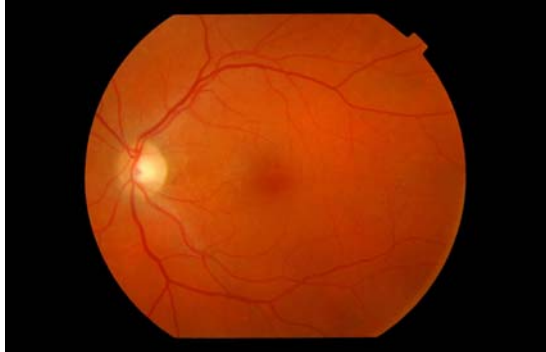
In Section 2 we outline our approach based on: position of the optic disc in the image, and assessment of image sharpness, and contrast. We present results based on these measures in Section 3 with a discussion of their discrimination in Section 4. Finally, conclusions and future work in quality assessment are given in Section 5.

2 Assessing image quality

The key elements in assessing the quality of retinal images are: optic disc detection and location, sharpness or focus measure, and contrast and brightness. Figure 1 shows four different left eye images. Both images (a) and (b) are of good quality; the first is fovea centred, while the second is optic disc centred. Image (c) illustrates an image of poor quality, in this case too blurry.

As can be seen in the sample images, Figure 1, the acquired image is rectangular, while the useful image data lies within a circular support F . All image processing techniques were restricted to this support.

Detection and location of the optic disc is important for determining if a sufficiently large enough field of view has been captured to enable detection of retinopathy. This requirement is called the 45° field of view (FOV) [4], Figure 2.



(a)



(b)



(c)

Figure 1: Various features of retina images from left eyes. (a) Fovea centred good image, (b) optic disc centred good image, and (c) Blurry image

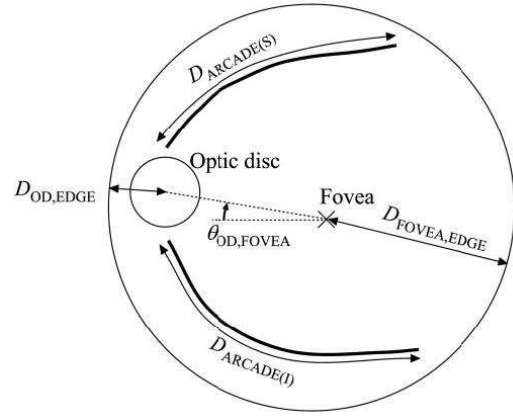


Figure 2: For a fovea centred image a 45° field of view will show the optic disc and sufficient quantity of the retina for assessment.

There are two requirements in order to meet the 45° field of view. Firstly, the full optic disc must be visible in the image. Secondly, for a fovea centred image approximately two disc diameters either side of fovea should be visible in the image. Alternatively, for an optic disc centred image approximately two disc diameters either side of the optic disc should be visible.

The optic disc is the circular area of maximal brightness in a retinal image[5]. Therefore the centre point of the optic disc is estimated by finding the largest cluster of pixels of the brightest portion of the image[6]. Once the centre point is estimated a region of interest is created, further processing for locating the optic disc is restricted to this region. Based on the work by Huang[7], the region of interest is ± 340 pixels from the estimated centre, see Figure 3.

This restriction is necessary because the boundary of the optic disc may be difficult to detect due to the large blood vessels crossing its boundary. Again following the work of Jelinek[5] and Huang[7] the perimeter is found through a process of applying a morphological closing operation to remove the blood vessels and noise, followed by a Canny edge detector [8, 9]. The parameters of the Canny edge detector are varied until a preset proportion, initially 0.29%, of pixels in the image are detected as edge pixels, Figure 4. From this estimates of the disc centre and radius are made.

If the radius of the optic disc is less than 100 pixels or greater than 200 pixels, then it is assumed that the optic disc is falsely detected [7]. The preset proportion of edge pixels is changed and the Canny edge detection process is reapplied. After the optic disc has been detected, the location of the optic disc is tested. If the centre estimate is near the centre of the image it is assumed to be optic disc

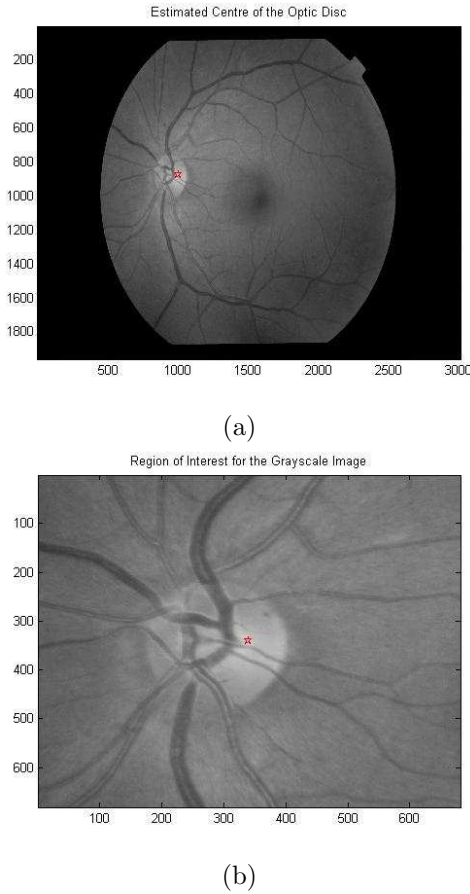


Figure 3: Finding the optic disc. (a) The red star is an estimate of the centre of the disc, based solely on image intensity. (b) A restricted region of interest centred on the estimate of (a).

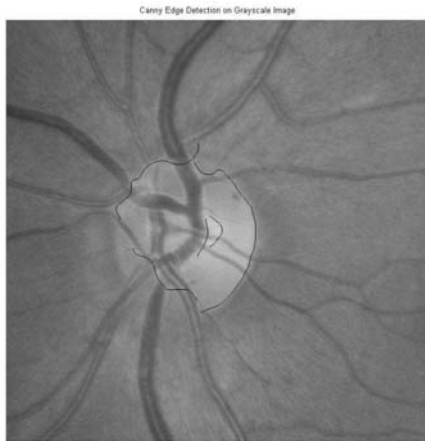


Figure 4: Superimposing the estimated optic disc perimeter back onto the source image.

centred. The circumference of a circle of diameter 2.5 times the disc diameter is tested to ensure all points lie within the captured support. Otherwise, the image is assumed fovea-centred and an arc of 4.5 times the disc-diameter is tested. To assist in this work an additional input, whether left or right eye, is also given to the program.

Clarity of the image is determined by examination of the sharpness, contrast and brightness of the image. Clarity is important, as grading is often based on the examination of very fine blood vessel structures in the retinal image. Image focus is the most important of these measures as image blurring can easily disguise lesions. Image blur occurs for a number of reasons, such as: poor optics, poor camera focus, lack of dialation, optical media opacities (cataracts) or patient movement during acquisition[4].

The proposed algorithm for measuring focus is based on a sharpness measure of the high frequency components of the image. Lower bandpass frequency components represent the slowly varying characteristics of an image, such as overall contrast and average intensity, whereas high frequency components characterise edges and other sharp details in an image [8]. Sharpness is defined as the ratio of the high frequency power to the bandpass power [10],

$$S = \int_{x,y \in F} \frac{HP^2(x,y)}{BP^2(x,y)} dx dy, \quad (1)$$

where $HP(x,y)$ is a measure of the high frequency, $BP(x,y)$ a measure of the bandpass frequency taken at points (x,y) that lie within the support (retina) of the image F .

The high and bandpass filters were implemented as separable IIR filters given by,

$$n^{IIR}(x) = \sum_{z=1}^N \alpha_z n(x-z) + \sum_{z=0}^M \beta_z m(x-z), \quad (2)$$

where $m()$ and $n()$ are the input and filtered output 1D image data respectively. The coefficients used for $BP(x)$ were $\alpha = \{-2.3741, 1.9294, -0.5321\}$, $\beta = \{0.0087, -0.0029, 0.0029, -0.0087\}$. The coefficients used for $HP(x)$ were $\alpha = \{0.0569, 0.35551, 0.03758\}$, $\beta = \{-0.0317, 0.0951, -0.0951, 0.0317\}$. The impulse responses for the filters are shown in Figure 5. As the blood vessels form only a small fraction of the image it is necessary for the HP filter to have a wide band and the largely uniform background a narrowband BP filter.

Histograms of the image intensities were also produced, together with measures of their mean value,

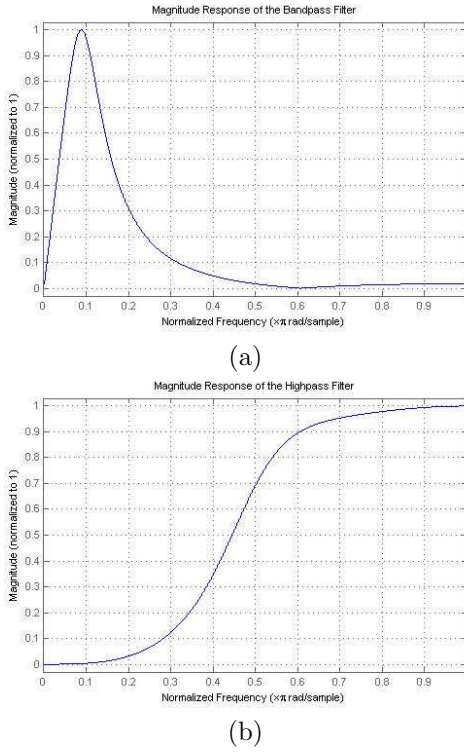


Figure 5: Impulse response of the sharpness filters. (a) The bandpass background filter, (b) the high-pass foreground filter.

standard deviation and skewness [8]. It is conjectured that images with large spread and therefore better contrast have better clarity.

3 Results

A sample of 100 test images were selected from the CDHB retinal image database and manually assessed for image quality and retinopathy. Table 1 shows the summarised results. Overall 72% of the images were of adequate quality, whereas the rest failed due to either poor field of view (FOV), insufficient clarity or other artifacts. Of these, 2 images had artifacts and 3 were graded as having both inadequate clarity and field of view.

Image Quality	FOV	Focus	Overall
Adequate	93	74	72
Inadequate	5	24	28

Table 1: Results of manually graded image quality.

These images were then automatically assessed for quality based on field of view and sharpness using a program written in MATLAB. In the case of the sharpness measure the image was 1:10 sub-sampled, for performance reasons, and an arbitrary threshold of 0.064 applied. If the overall sharpness measure is less than or equal to 0.064, then the image is classified as blurred, otherwise the

image is classified as in focus. Table 2 contains the summarised result obtained from the designed automated system.

Image Quality	FOV	Focus	Overall
Adequate	67	68	46
Inadequate	33	32	54

Table 2: Results of automatically graded image quality.

These results show that a significantly lower percentage of images, 46%, were classified as having adequate quality. Of the failed images 11 were rejected on both grounds of field of view and focus. However, the images rejected by the manual grader for artifacts passed the quality tests used here.

The low acceptance rate is not-necessarily a problem, given the initial aim of producing a sub-database of images for further image analysis. More importantly is the measure of false-positives and false-negatives. It is highly desirable that the false-positive rate is near zero, Table 3.

Image Quality	FOV	Focus	Overall
False Positives	3:67	5:68	5:46
False Negatives	31:33	13:32	31:54

Table 3: Measures of false-positive and false-negative classifications.

A preliminary sensitivity analysis of the sharpness threshold was also conducted. Table 4 shows the percentage false-positive and false-negative rate for focus classification as the threshold value is changed.

Threshold Value	0.060	0.064	0.065
False Positives (%)	2	5	7
False Negatives (%)	32	13	8

Table 4: False detection rates for various setting of the sharpness threshold.

4 Discussion

The computed results of the automated quality classification system raise some interesting points for discussion. The false-positive and false-negative classifications are high. Having an inadequate quality image incorrectly graded as being of adequate quality (i.e. false-positive detection) is undesired, as images that are supposed to be rejected are used for further analysis. False-negative detections are less troublesome, only the efficiency of the automated

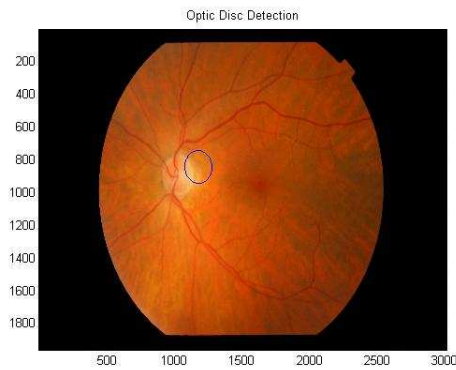


Figure 6: Example of unsuccessful optic disc detection.

system is decreased because the good quality pictures are unnecessarily discarded.

There are a number of physiological factors that might influence the false detection rate, including: the increased reflectivity of a young person's retinal image and pigmentation. Here we concentrate more on algorithmic approaches that influence these false detection rates, in turn suggesting possible changes to improve our performance.

Our assessment of the adequacy of the field of view is determined by following the approach of Huang[7] for optic disc detection. This method has the appeal of simplicity. It is, however, reliant on adequate detection of the disc perimeter in order to produce a safe estimate of the disc centre and diameter. Problems, such as triangulation errors from edge estimates too close together must be avoided. These typically lead to false-negative errors. Whereas an erroneously computed perimeter could lead to seemingly sensible but wholly wrong disc parameters, leading to false-positive classifications, Figure 6.

Jelinek et. al.[5] suggest that the low contrast of the background of the image and the slow variation of the optic disc to the background does not help in discerning the optic disc perimeter. They recommend some statistical technique to help enhance the optic disc detection.

Alternative optic disc and field of view methods include the work by Foracchia et. al.[11], where they reported an error rate of 98%. In this approach the main retinal vessels (arcades) are identified and used to locate the disc. Fleming et. al.[4] also follow this approach using the generalised Hough transform to find the semi-elliptical shaped arcades. In Lalonde, et. al.[12], pyramidal decomposition and Hausdorff-based template matching technique are used to detect the optic disc, where an average error of 7% on optic disc centre positioning was reached with no false de-

tection. Another method can be found in Li and Chutatape[6], again based on template matching and Principal Component Analyses (PCA). A comparative analysis of these different techniques is warranted.

The sharpness measure applied in this work is, unfortunately, not very discriminatory, producing a significant number of false-positives and false-negatives. As can be seen in Table 4 the performance is also significantly sensitive to the chosen threshold value. An issue here may be that the images were not normalised in image power. The method described in equation (1) attempts to compensate for this by measuring image power through the use of the bandpass filter, however too many low frequency power terms may have been discarded.

An alternative sharpness method considered was that described by Dijk, et. al.[13]. This method was initially rejected due to the simplicity of the algorithm from Shaked and Tastl[10]. Fleming et. al.[4] looked at image clarity in another way. Their method is based on an algorithm to measure the total length of blood vessels in the image. An image is classified as having adequate clarity if this measure exceeded a threshold determined from a training set. Their method is based on the concept that only the finer blood vessels can be measured if the image has sufficient focus and contrast. Closer examination of this work is warranted.

Our preliminary examination of the effectiveness of the image intensity histogram and colour measures as a means of measuring image quality has, as yet, proven not to be insightful. These remain areas for us to continue our work.

The establishment of a database of gold standard assessed images will also help improve the statistical analysis of this work. At this stage no such national database exists, but work is underway to do so by Save Sight Society and the Ministry of Health.

5 Conclusion and future work

The burgeoning epidemic of diabetes in New Zealand, and the western nations in general, threatens to overwhelm health budgets, as much as 20% in New Zealand [14]. It is one of the leading causes of blindness [1]. The effective treatment of diabetic retinopathy is reliant on a constant monitoring scheme of the patient and an effective screening programme. However, the sheer number of images requiring reading could easily exceed the manual resources currently used for this grading purpose. This paper described preliminary work to assess retinal image quality

as a pre-cursor to subsequent image analysis and a computer assisted screening programme.

The results obtained, based on field of view and image sharpness, are a good initial step. However improvements both in terms of false-positive classification and efficiency (false-negatives) are needed. Potential improvements were discussed with the greatest effort required in improving the estimate of image clarity.

In assessing the results we came across the work by Fleming, et. al.[4]. We have followed a similar approach, the difference lying principally in the measure of image clarity. Their technique is based on identification and measurement of the blood vessels in the image. We have reservations about the applicability of this work to images showing proliferative diabetic retinopathy. We also have reservations about the speed and computation load of their algorithm. However, it is an interesting approach and important for us to critically analyse its performance.

There is a possibility that this image quality assessment algorithm could also be used during the time of photographing the patients. Potentially photographers could employ such advancements to assess image quality, rejecting inadequate images and immediately re-taking the image.

The computational load of the algorithm has not been explicitly examined at this stage. While this will be important for processing the current CDHB retinal image database of 100,000 images, in the context of a photographer assisted system it is largely irrelevant, as the processing time is significantly small compared to the acquisition time.

References

- [1] N. Z. M. of Health, "Diabetes in new zealand: Models and forecasts 1996-2011," tech. rep., New Zealand Ministry of Health, 2002.
- [2] H. Li and O. Chutatape, "Fundus image features extraction," in *Proceedings of the 22nd Annual EMBS International Conference*, pp. 3071–3073, July 2000.
- [3] S. S. S. of New Zealand Ltd., "National diabetes retinal screening grading system and erferral recommendations 2005," tech. rep., Save Sight Society of New Zealand Ltd., 2005.
- [4] A. Fleming, S. Philip, K. Goatman, and J. O. P. Sharp, "Automated assessment of diabetic retinal image quality based on clarity and field definition," *Investigative Ophthalmology and Visual Science*, no. 3, pp. 1120–1125, 2006.
- [5] H. Jelinek, C. Depardieu, C. Lucas, D. Cornforth, W. Huang, and M. Cree, "Towards vessel characterisation in the vicinity of the disc in digital retinal images," in *Proceedings Images and Vision Computing New Zealand 2005*, pp. 351–356, November 2005.
- [6] H. Li and O. Chutatape, "Automatic location of optic disk in retinal images," in *IEEE International Conference on Image Processing 2001*, pp. 837–839, August 2001.
- [7] W. Huang, "Automatic detection and quantification of blood vessels in the vicinity of the optic disc in digital retinal images," 2006.
- [8] R. Gonzalez and R. Woods, *Digital Image Processing*. Massachusetts: Addison-Wesley, 1992.
- [9] R. Fisher, S. Perkins, A. Walker, and E. Wolfart, "Canny edge detector." <http://homepages.inf.ed.ac.uk/rbf/HIPR2/canny.htm>, accessed 3 September 2006.
- [10] D. Shaked and I. Tastl, "Sharpness measure, towards automatic image enhancement," in *IEEE International Conference on Image Processing 2005*, pp. 937–940, September 2005.
- [11] M. Foracchia, E. Grisan, and A. Ruggeri, "Detection of optic disc in retinal images by means of a geometrical model of vessel structure," *IEEE Transactions on Medical Imaging*, no. ?, pp. 1189–1196, 2004.
- [12] M. Lalonde, M. Beaulieu, and L. Gagnon, "Fast and robust optic disc detection using pyramidal decomposition and hausdorff-based template matching," *IEEE Transactions on Medical Imaging*, no. ?, pp. 1193–1200, 2001.
- [13] J. Dijk, M. van Ginkel, R. van Asselt, L. van Vliet, and P. Verbeek, "A new sharpness measure based on gaussian lines and edges," in *8th Annual Conference of the Advanced School for Computing and Imaging*, pp. 39–43, June 2001.
- [14] PricewaterhouseCoopers, "Type 2 diabetes: Managing for better health outcomes. economic report for diabetes new zealand inc.," tech. rep., PricewaterhouseCoopers Ltd., 2006.

Results of a multiple-baseline interferometric synthetic aperture sonar in shallow water

M. P. Hayes

²University of Canterbury, Dept. Electrical and Computer Engineering.

Email: m.hayes@elec.canterbury.ac.nz

Abstract

This paper presents preliminary results obtained with KiwiSAS-4, an experimental multiple-baseline interferometric synthetic aperture sonar (InSAS). The sonar can be configured as a three element vertical interferometer. Super-resolution techniques are applied to experimental data obtained in the shallow waters of Lyttelton Harbour. The results show a marginal improvement of bathymetry can be obtained using the additional element to discriminate between the direct and sea-surface multipath echoes.

Keywords: InSAS, bathymetry, multipath, multibaseline

1 Introduction

KiwiSAS-4 is an experimental Interferometric Synthetic Aperture Sonar (InSAS) developed by the Acoustics Research Group at the University of Canterbury. Essentially it is a reconstruction of KiwiSAS-3 [1], modified to allow the signals measured by the nine Polyvinyl Difluoride (PVDF) tiles comprising the hydrophone array to be individually recorded. These hydrophones are arranged as a three by three matrix and can be configured to act as a three element vertical interferometer. Like KiwiSAS-3, KiwiSAS-4 operates in two simultaneous frequency bands (20–40 kHz and 90–110 kHz) using the same sets of transducers.

One of the motivations of the sonar is to determine whether some of the multipath signals in a shallow water environment can be suppressed using the additional transducers. These multipath signals can introduce bathymetric artefacts since they violate the assumption that there is a single scatterer in each range resolution cell. Spatially resolving the scatterers with a large array is infeasible, especially at low frequencies, due to the vertical space constraints of a towfish or Autonomous Underwater Vehicle (AUV). Thus super-resolution techniques are required.

This paper starts with a review of the techniques used for standard interferometric bathymetry and how they can be extended to multiple-baseline interferometric bathymetry. Then there is a brief review of the multipath problem before a presentation of experimental results obtained from sea trials.

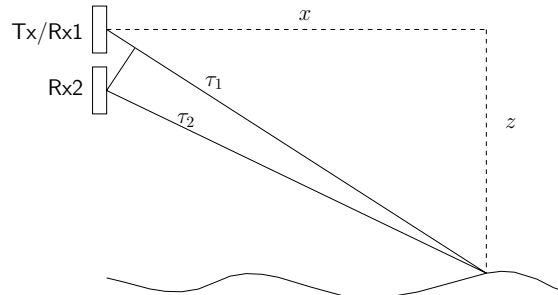


Figure 1: Two element interferometer geometry. Tx/Rx1 denotes the transmitter collocated with receiver 1, Rx2 denotes receiver 2. x and z are estimated from measurements of τ_1 and τ_2 .

2 Interferometric sidescan sonar

Bathymetric imaging is an inverse problem where the seafloor topography is estimated from the echoes recorded by a sonar. With a standard interferometric sidescan sonar, two vertically displaced hydrophones are employed as shown in Figure 1. The goal is to estimate the across-track position x and depth z relative to the sonar for each imaged point on the seafloor. These measurements are estimated by correlating the received echoes. For example, consider a projector at $(x_p, 0, z_p)$, a pair of hydrophones at $(x_{h1}, 0, z_{h1})$ and $(x_{h2}, 0, z_{h2})$, and an isolated scatterer at $(x, 0, z)$. The propagation delay as a function of scatterer position for the n^{th} hydrophone is

$$\tau_n(x, z) = \frac{1}{c} \sqrt{(x - x_p)^2 + (z - z_p)^2} + \frac{1}{c} \sqrt{(x - x_{hn})^2 + (z - z_{hn})^2}, \quad (1)$$

where c is the speed of sound (assuming an iso-velocity profile). Denoting the pulse compressed

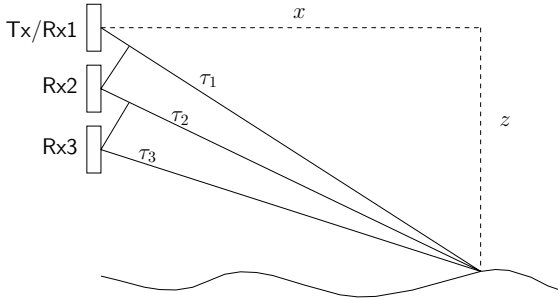


Figure 2: Three element interferometer geometry.

signals recorded at the two hydrophones by $d_1(t)$ and $d_2(t)$, the bathymetric problem is to estimate the seafloor height $z(x)$ as a function of across-track position x . Assuming that $z(x)$ is single-valued, one approach is to maximise the correlation between the two signals for each desired across-track position x by searching over the expected range of seafloor heights z . Mathematically this can be described by

$$\hat{z}(x) = \arg \max_z \left| \int_{-\infty}^{\infty} \chi_{12}(t, \tau_1, \tau_2) dt \right|, \quad (2)$$

where

$$\begin{aligned} \chi_{12}(t, \tau_1, \tau_2) = & d_2(t - \tau_2) d_1^*(t - \tau_1) \\ & \times \text{rect} \left(\frac{t - \tau_1}{T} \right) \text{rect} \left(\frac{t - \tau_2}{T} \right), \end{aligned} \quad (3)$$

and where T is the extent of a time observation window.

With the assumption that there is a single scatterer, the bathymetric problem is equivalent to a time delay estimation problem. This can be extended to multiple scatterers, provided there is only a single dominant scatterer in any range resolution cell, by estimating the time delay between the two pulse-compressed signals averaged over an interval T at each delay τ of interest, i.e.,

$$\hat{\Delta\tau}(\tau) = \arg \max_{\Delta\tau} \left| \int_{-\infty}^{\infty} \chi_{12}(t, \Delta t, \tau) dt \right|, \quad (4)$$

where

$$\begin{aligned} \chi_{12}(t, \Delta t, \tau) = & d_2(t + \Delta\tau) d_1^*(t) \\ & \times \text{rect} \left(\frac{t - \tau}{T} \right) \text{rect} \left(\frac{t - \tau - \Delta\tau}{T} \right). \end{aligned} \quad (5)$$

The differential delay estimates $\hat{\Delta\tau}(\tau)$ are then mapped to seafloor height estimates $\hat{z}(x)$ using simple geometry.

The variance of the differential time delay estimates about the true delays can be determined from time estimation theory. The Cramér-Rao lower bound (CRLB) is inversely proportional to the observation time T [2] and thus there is a trade-off between resolution and height accuracy. When there are multiple hydrophones, see for example Figure 2, the estimate of z can be improved by summing the (baseline time-scaled) correlations between the signals, inversely weighted by the expected variances. The variance of the estimate of z can be further reduced by averaging the correlations over neighbouring along-track positions (multilook processing) although this reduces the along-track resolution.

The time delay estimation can be performed in either the time or frequency domains; the latter usually using a number of frequency bands to allow narrowband approximations. When the signals are narrowband (as is common in interferometric radar) the correlations can be simplified to a Hermitian product. The phase of the Hermitian product is called an interferogram and is proportional to the interferometric time delay and centre frequency of the frequency band. However, with any narrowband time delay estimation technique, the time delay estimate is ambiguous and requires unwrapping. The unambiguous time delay interval can be extended using a lower frequency (but with a degradation in accuracy) or by employing additional hydrophones (multiple baseline) [3]. It can also be extended by increasing the signal bandwidth [4]. Once the signal bandwidth is comparable with half the centre frequency, ambiguities can be avoided and thus phase unwrapping is unnecessary. However, employing narrowband time delay estimation techniques with broadband signals results in a decorrelation due to the footprint shift effect [5].

3 Multipath

Shallow water sidescan sonar imagery is corrupted by unwanted reflections from the sea-surface, a phenomenon known as multipath. Ideally the acoustic energy travels from the projector (transmitter) to the seafloor and the scattered energy travels back to the hydrophone (receiver). However, some of the scattered energy will be reflected from the sea-surface and be received at the hydrophone a short time later. Moreover, some of the transmitted acoustic energy will be reflected from the sea-surface before being scattered from the seafloor as illustrated in Figure 3. The echo signals due to these additional paths are called multipath echoes.

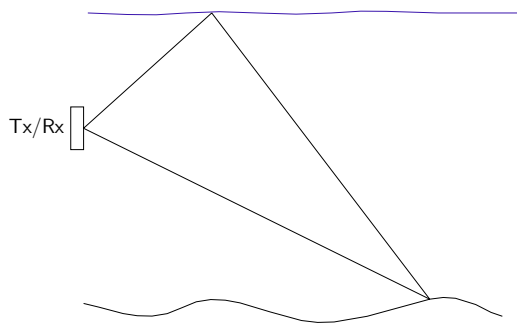


Figure 3: Illustration of the first-order multipaths due to reflections from the sea-surface. As well as the direct path from the sonar to the seafloor, the transmitted and received signals can be reflected from the sea-surface.

By careful control of the projector vertical beam pattern, the transmitted energy that is reflected from the sea-surface can be minimised. However, it is not straightforward to minimise the seafloor scattered echoes that are reflected off the sea-surface. While in theory it is possible to differentiate these echoes from the direct echoes, on the basis of angle of arrival, this is difficult at low frequencies due to the vertical space constraints of a towfish. Thus super-resolution techniques are required. Unfortunately, multipath echoes violate the assumption of standard interferometric techniques that there is only a single plane wave incident upon the hydrophones in any range resolution cell.

Resolving multiple echoes within a resolution cell requires a more sophisticated echo model, additional elements in the interferometer, and a multidimensional search. Performing a direct multidimensional search is prohibitively expensive since the search space is non-convex [6]. A more promising approach has been to replace the multidimensional search with an iterated single dimensional search. This technique is called the RELAX algorithm [7] and has been applied to InSAR layover estimation and InSAS multipath estimation [8, 9]. Unfortunately, this technique cannot be applied to higher order multipath components that are reflected off the sea-surface and again off the seafloor since they have a similar arrival angle (and thus interferometric delay) to the direct echoes.

4 Results

The results in this paper were acquired from a sea trial of the KiwiSAS-4 sonar in late January 2006. The sea trial was conducted in Lyttelton Harbour, the major port of the South Island of New Zealand, in the vicinity of Parson’s Rock—a basalt protrusion 5 m proud of the harbour mud.

The neutrally buoyant KiwiSAS-4 towfish was nose-towed with a depressor chain at speeds in the range of 1–2 m/s past Parson’s Rock at a nominal depth of 5 m with a water depth of 10–12 m. Linear FM chirps were simultaneously transmitted between 20–40 kHz and 90–110 kHz for a duration of 12.5 ms at a repetition frequency of 15 Hz. The projector was steered down by 12° in an attempt to reduce the multipath signals resulting from the transmitted signal being reflected from the sea-surface.

On the day of the trial the sea was remarkably smooth for a summer’s day and this gave rise to noticeable sea-surface multipath as shown in Figure 4. These images were reconstructed using a wavenumber reconstruction algorithm for each frequency band and hydrophone array. Note that the multipath (the ghosting in the across-track direction) is less visible in the 100 kHz images, primarily due the narrower vertical beam patterns of the hydrophones.

The coherence between the top and middle hydrophone arrays is shown in Figure 5. This was calculated over a small frequency band centred at 30 kHz and 100 kHz respectively. The coherence is much higher at lower frequencies and in the region of Parson’s Rock. It is envisaged that this is due to volume scattering effects within the layer of fine suspended sediment above the seafloor mud and to a lower backscatter coefficient.

The narrowband interferometric phase is shown in Figure 6. This shows a steadily increasing slope up to the top right hand corner. This is consistent with depth-sounding measurements of the area.

The RELAX algorithm [9] was then applied using 9 frequency subbands and 5 along-track looks to the 30 kHz data. The results are shown in Figure 7(a) and Figure 8(a) for the single target hypothesis (equivalent to a standard multiple-baseline ML estimator). Figure 7(b) and Figure 8(b) show the results for the component of the dual target hypothesis that falls within the expected seafloor height range. While both images show artefacts where the coherence is low, there is a slight improvement when two components are estimated. This is more noticeable in the 30–35 m across-track region.

A surprising result is that most of the multipath energy appears to be arriving from below the sonar, rather than from a sea-surface reflection as would be expected with tilting of the projector toward the seafloor. Thus the dual target hypothesis has difficulty resolving these multipath signals from the direct path signals since they are at comparable arrival angles. This is confirmed by the comparing the amplitude images Figure 8(a) and Figure 8(b)

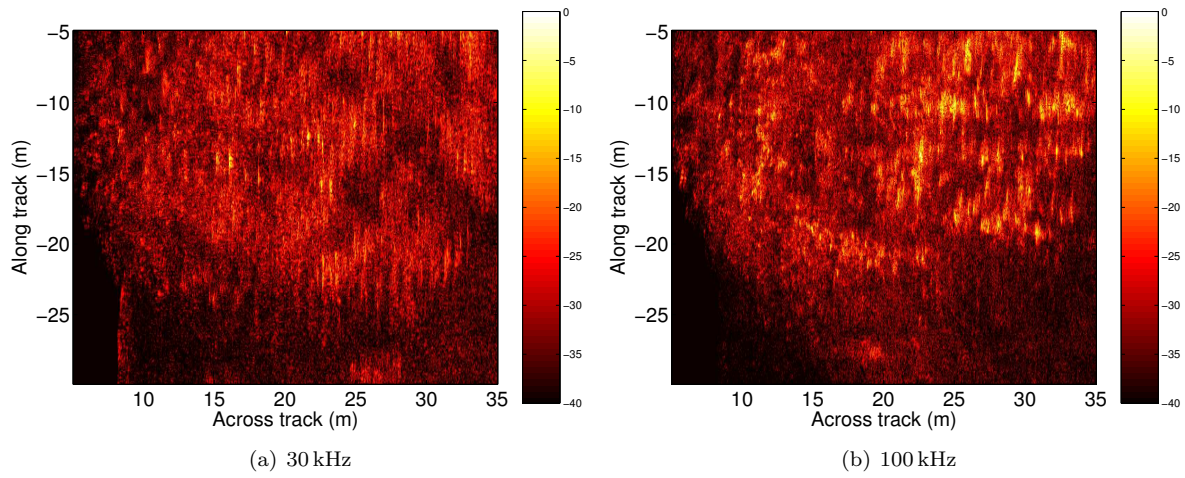


Figure 4: Reconstructed SAS images for middle hydrophones at 30 kHz and 100 kHz.

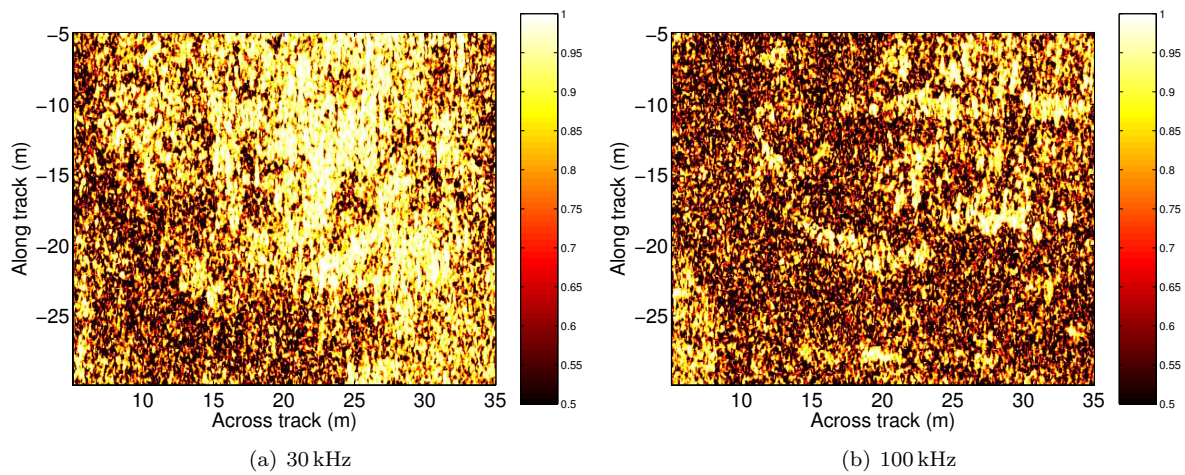


Figure 5: Estimated coherence between top and middle hydrophones at 30 kHz and 100 kHz.

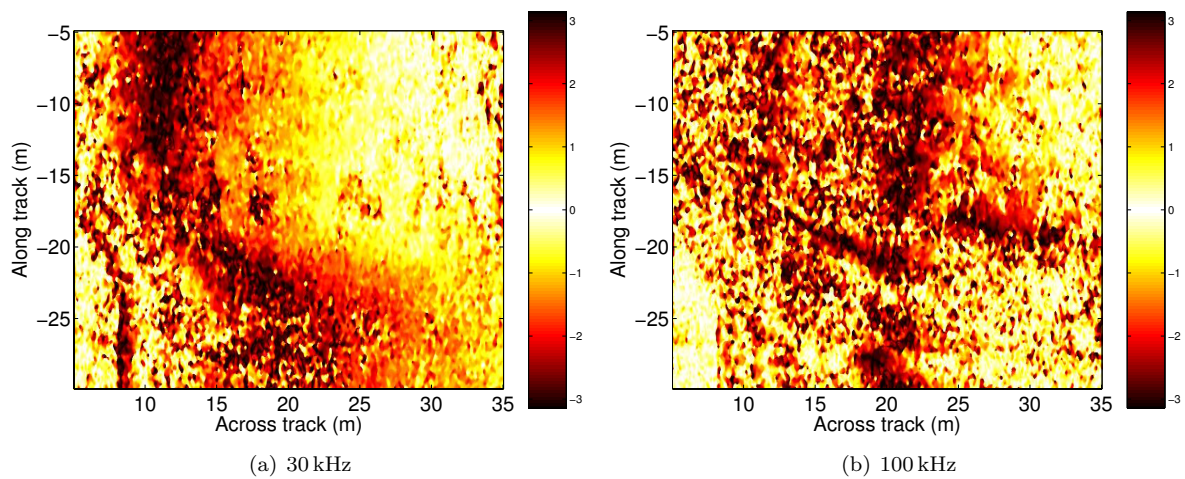


Figure 6: Estimated interferometric phase (median filtered) between top and middle hydrophones at 30 kHz and 100 kHz.

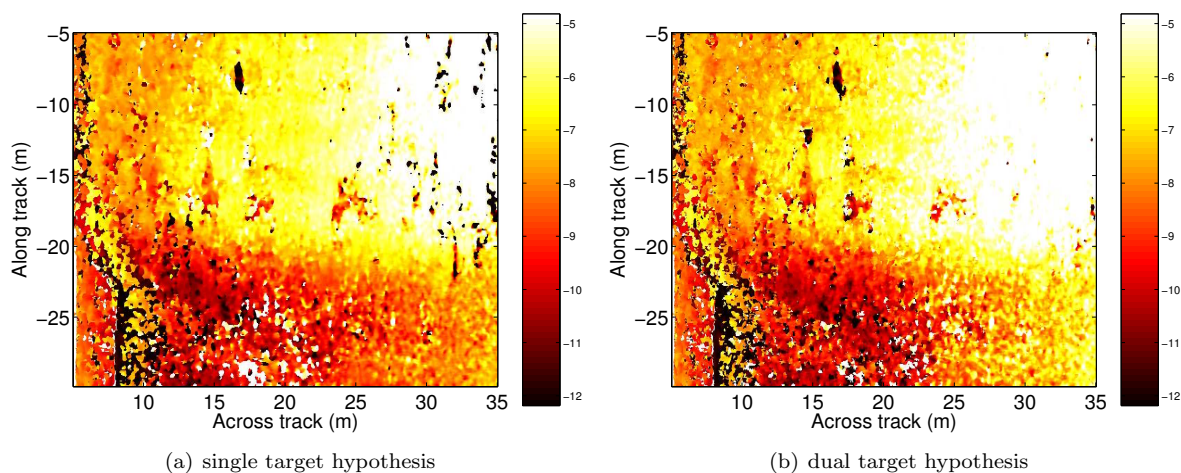


Figure 7: Estimated bathymetric images using RELAX algorithm for single target and dual target hypotheses with three elements.

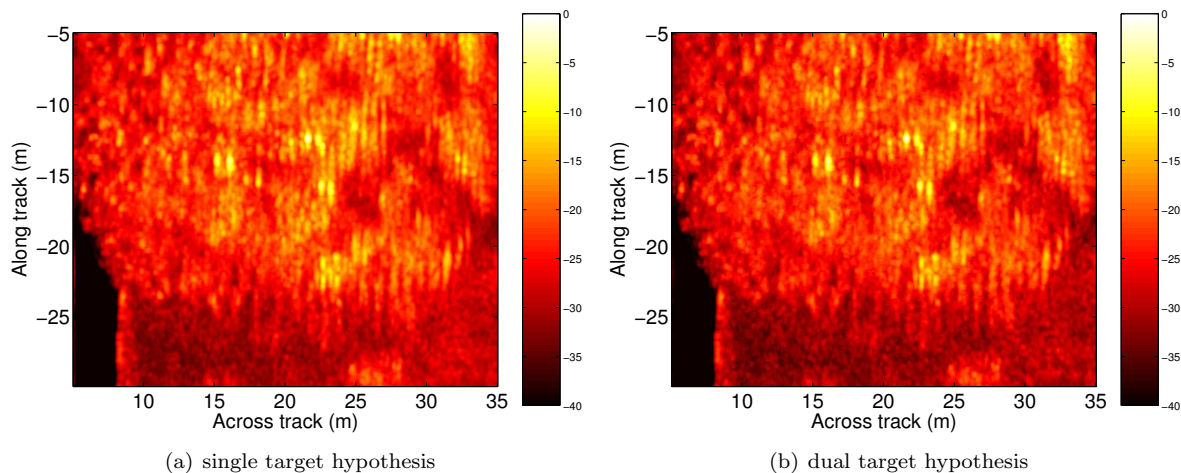


Figure 8: Estimated amplitude (dB) images using RELAX algorithm for single target and dual target hypotheses with three elements.

for the single and dual target hypotheses. This shows little reduction of the ghosting artefacts.

5 Conclusions

Interferometric imaging in shallow waters suffers from sea-surface multipath. The multipath echoes sometimes violate the interferometric assumption that there is only a single echo arrival at each range. The RELAX algorithm did not give a great improvement in bathymetric image quality in regions where it appears that there is significant sea-surface multipath. One of the reasons appears to be violation of the assumption that the echo amplitude is the same for each hydrophone [9]. This is due to vertical beam-pattern effects (and slight uncompensated differences in hydrophone sensitivity). Further work is required to adapt the estima-

tion algorithm to model the vertical beam-pattern response.

6 Acknowledgements

The author thanks Prof. Peter Gough and Mike Cusdin for their assistance collecting the data presented in this paper.

References

- [1] M. P. Hayes, P. J. Barclay, P. T. Gough, and H. J. Callow, "Test results from a multi-frequency bathymetric synthetic aperture sonar," in *Oceans 2001*, (Honolulu, Hawaii), pp. 1682–1687, November 2001.
- [2] A. H. Quazi, "An overview on the time delay estimate in active and passive systems for target localization," *IEEE Trans. Acoustics*,

Speech and Signal Processing, vol. 29, pp. 527–533, June 1981.

- [3] G. Corsini, M. Diani, F. Lombardini, and G. Pinelli, “Simulated analysis and optimization of a three-antenna airborne InSAR system for topographic mapping,” *IEEE Trans. on Geoscience and Remote Sensing*, vol. 37, pp. 2518–2529, Sept. 1999.
- [4] R. Lanari, G. Fornaro, D. Riccio, M. Migliaccio, K. P. Papathanassiou, J. R. Moreira, M. Schwäbisch, L. Dutra, G. Puglisi, G. Franceschetti, and M. Coltelli, “Generation of digital elevation models by using SIR-C/X-SAR multifrequency two-pass interferometry: The Etna case study,” *IEEE Trans. on Geoscience and Remote Sensing*, vol. 34, pp. 1097–1114, Sept. 1996.
- [5] X. Lurton, “Swath bathymetry using phase difference: theoretical analysis of acoustical measurement precision,” *IEEE J. Oc. Eng.*, vol. 25, pp. 351–363, July 2000.
- [6] F. Lombardini, M. Montanari, and F. Gini, “Reflectivity estimation for multibaseline interferometric radar imaging of layover extended sources,” *IEEE Trans. Signal Processing*, vol. 51, pp. 1508–1519, June 2003.
- [7] J. Li and P. Stoica, “Efficient mixed-spectrum estimation with applications to target feature extraction,” *IEEE Trans. Signal Processing*, vol. 44, pp. 281–295, Feb. 1999.
- [8] M. P. Hayes, “Multipath reduction with a three element interferometric synthetic aperture sonar,” in *Proceedings of the Seventh European Conference on Underwater Acoustics*, (Delft, The Netherlands), pp. 1151–1156, ECUA2004, July 2004.
- [9] M. P. Hayes, A. J. Hunter, P. J. Barclay, and P. T. Gough, “Estimating layover in broadband synthetic aperture sonar bathymetry,” in *Oceans 2005 Europe*, (Brest, France), IEEE/MTS, July 2005.

Monocular tracking of swimmers from a stationary viewpoint

C.P. Huynh and R. Green

Department of Computer Science & Software Engineering
University of Canterbury
Private Bag 4800 Christchurch, New Zealand
Email: cph35@student.canterbury.ac.nz
Email: richard.green@canterbury.ac.nz

Abstract

This paper proposes a method to track the motion of swimmers through the noisy background of rippling water. The method we employ is a combination of an adaptive background subtraction algorithm, a binary blob filtering algorithm, and a statistical analysis on the temporal change in the area and dimension of the resulting blobs. Subsequently, we evaluate the success rate of adapting this method to function in the noisy swimming pool background. Our experimental results show that the areas and dimensions of segmented binary foreground blobs can be used to track swimmers' positions, but not sufficient for detecting cyclic motion. Techniques such as colour-based segmentation and parameterization of human body parts are proposed for further research to segment and detect cyclic human motion in this noisy environment.

Keywords: Tracking, cyclic motion detection, background subtraction, blob filtering.

1 Introduction

Tracking human motion is an important research field in Computer Vision, which can be seen as a separate process, or as a means to prepare data for human pose estimation and recognition. Moeslund [1] provides a comprehensive survey on human motion tracking techniques existing in the literature. Specifically, the cyclic motion of a walking figure gives cues to identity because it encodes several human body characteristics such as stride, height and frequency, on which classification and recognition could be performed.

In this paper, we develop a tracking technique using similar ideas to human gait analysis and experiment it on image sequences of swimmers. To date, most of the research on tracking cyclic motion assumes a static background, and does not focus on dealing with moving background elements and noisy background. The main motivation of this study is to explore and evaluate tracking techniques applied to environments with noisy background such as swimming pools. Findings of the shortcomings of our techniques are reported, based on which alternative or complementary methods are suggested.

This paper is organized as follows. In section 2, a background of the relevant research on gait analysis is presented. Section 3 describes our

tracking method as a three-phased process. Section 4 shows the experimental conditions and the results for four swimming styles: freestyle, backstroke, breaststroke, and butterfly. Section 5 discusses the weaknesses of the method and suggests improvements. The last section includes a conclusion and future work.

2 Related work

Tracking techniques can be categorized as object-based and image-based regarding the data representation. Object-based approaches are based on foreground segmentation while image-based approaches derive information directly from the image.

An early example of image-based approaches is the study in [2] on cyclic motion detection by considering the 2-D trajectory of a single point on a moving object. The trajectory is represented as a spatio-temporal curve in the (x, y, t) space. Performing autocorrelation and Fourier transforms on a smoothed version of this curve results in a Fourier plot, in which large impulses correspond to a cyclic frequency. However, tracking points requires the attachment of markers to the tracked objects, but it is awkward and infeasible to do so to objects such as swimmers. Furthermore, points

do not provide as rich information as the temporal change of the whole body volume.

In a number of tracking methods, it is common to model the human body as a collection of limbs to form a connected kinematic tree [3]. In repetitive human motion, the motion sequences decompose into similar motion cycles. Using the human body model, angles between different body parts are tracked because they indicate the phase of cyclic motion. In addition, Principle Component Analysis (PCA) models of human gaits also become quite standard. Urtasun [4] represents the body of a golfer as a set of volumetric primitives attached to a 3-D articulated skeleton.

In addition, due to its simplicity, silhouette representation is also popular and used in [5] [6] [7] to detect the cycles of walking figures. Silhouette can be obtained by background subtraction or thresholding in figure-ground segmentation.

3 Method

We use a static high resolution camera mounted at the side of the swimming pool, which looks down the target lanes, to capture video footages of swimmers. Image sequences of swimmers are then processed through the following phases, as shown in figure 1.

3.1 Background modelling and foreground segmentation

A statistical background model is approximated and the foreground image of the target swimmer is segmented according to the Adaptive Mixture of Gaussian background subtraction method [8]. We use this background subtraction technique because its background model adapts effectively to lighting changes, repetitive motions, and long-term scene changes. This algorithm models each background pixel by a mixture of K Gaussian distributions (where K is a small number from 3 to 5).

The probability that a pixel has a value of I at time t is

$$P(I_t) = \sum_{i=1}^K \omega_{i,t} \eta(I_t; \mu_{i,t}, \Sigma_{i,t})$$

where $\omega_{i,t}$ is the weight parameter of the i^{th} Gaussian component and $\eta(I_t, \mu_{i,t}, \Sigma_{i,t})$ is the normal distribution of that component, with $\mu_{i,t}$ as the mean and $\Sigma_{i,t}$ as the covariance of the i^{th} component.

The K components are sorted into the decreasing order of the fitness value $\frac{\omega_{i,t}}{|\sum_{i,t}|}$. The higher the ratio the more likely the component is part of the

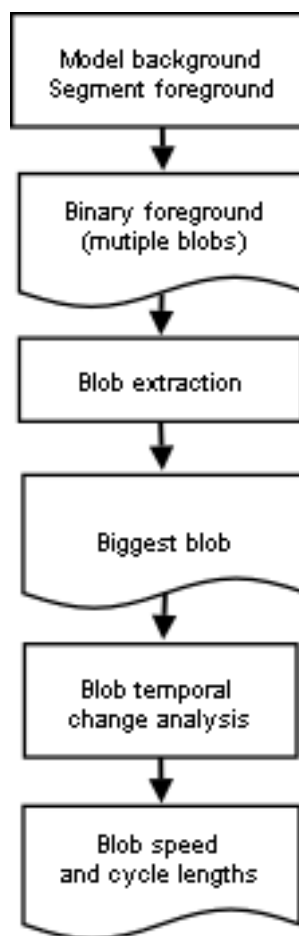


Figure 1: Calculating cyclic motion frequency.

background. Given a threshold T , the first B components are used as a model of the background where B is estimated as

$$B = \operatorname{argmin}_b \left(\frac{\sum_{i=1}^b \omega_{i,t}}{\sum_{i=1}^K \omega_{i,t}} > T \right)$$

The other important parameter of this model is its learning rate, α , which determines how fast it is for a Gaussian component to be included as part of the background.

According to [9], only two parameters α and T need to be set for the system. From our experiments with different parameter values of the background model, we observe that with $T = 0.7$ and $\alpha = 0.005$, we can filter out considerable more noise blobs from the original video footage compared to other parameter value configurations.

Figure 2 shows the original image of a swimmer in a noisy background with splashes and light reflection from the water, and the foreground image segmented from the original using the Adaptive Mixture of Gaussian algorithm.

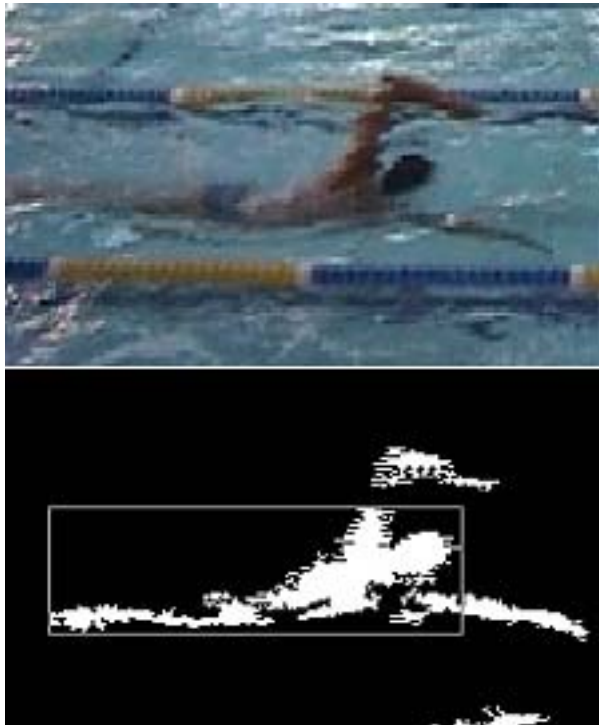


Figure 2: Original image of a swimmer and the segmented binary foreground image.

3.2 Blob extraction

The foreground image resulting from the previous phase contains several white pixel blobs, one of which is the moving swimmer blob. This blob is then extracted from the binary image. To increase the chance of correctly identifying the moving blob, we have the following assumptions about the test environment. One is that swimmers only move in a single direction throughout the whole image sequence. Therefore, any blobs whose horizontal coordinates are out of order compared to the swimmer's positions in the previous and subsequent frame are not considered as a candidate swimmer blob. Secondly, for simplicity, we only track a swimmer. Therefore we can choose to process only the region of interest corresponding to the swimmer's lane. Thirdly, provided that there are no big water splashes, noise blobs are usually smaller than the swimmer blob. Therefore, the largest blob within the target lane is assumed to be the swimmer blob.

3.3 Speed and cycle length calculation

The swimming speed is calculated as the horizontal difference between the blob positions in successive frames. The blob positions in successive frames are assumed to be strictly increasing or decreasing provided there is no noise. If the horizontal position of a swimmer blob is out of order compared to

its preceding and succeeding frames, it is simply discarded from the statistics.

An important cue to determine the cyclic motion frequency of a swimmer is the temporal changes of the binary swimmer blob. During a full motion cycle, it is intuitive that the swimmer blob attains its maximum area when the swimmer's arms enter the water. This is because commonly the largest blob seen at this stage is a combination of the swimmer and the resulting splashes. In addition, for a number of swimming styles, such as butterfly, the swimmer's body stretches furthest at this stage. Therefore, we can also examine the variation of the width of the swimmer blob to determine when the swimmer's arms enter the water. Similarly, the variation of the blob's height can also be examined for the same purpose.

The width variation of the blob can be expressed as a single-valued function of time (frame number). By examining the graph of this function, we can identify successive local maximal values, which are approximately a full motion cycle apart in time. A point on the graph is a local maximum if its value is greater than a certain percentage p of points in its neighbourhood. Currently, the neighbourhood is modelled as a sliding window moving along the graph. When a local maxima is detected, the window size is adjusted to be a proportion q of the length of the last cycle and the window is shifted by a proportion r of the cycle length. We experiment with different values of these parameters to obtain an accurate configuration for finding the local maximal blob widths.

4 Experiments and results

We conduct our experiments on several swimming styles: freestyle, backstroke, breaststroke and butterfly. The inputs to our experiments are 25 frames per second videos of swimmers in uncontrolled background conditions: bright lights and high frequency water ripples. A single combination of parameter values is used in all the experiments.

Figure 3 shows the calculated swimming speed in different styles. The general trend is that the speeds estimated in the middle of an image sequence tend to be more stable than that at the beginning and the end of the sequence. This is explained by the fact that the swimmer's body length is a main factor affecting the choice of the largest blob of motion, and the images at the beginning and the end of the sequence do not contain the whole swimmer's body. Very high speeds occasionally occur in the middle of an image sequence due to the incorrect identification

of the moving blob. This is because at times the segmented swimmer's blob might be smaller than a water ripple or splash blob. For simplicity, we discard all these noisy data points when averaging the swimmer's speed over a period of time.

The width of segmented swimmer's blob does not strictly vary in a periodical pattern over time as shown in figure 4. Noise blobs, including water ripples and splashes, are often mistaken for the swimmer's blob or combined with the swimmer's blob into a bigger one. As a consequence, the presence of these noise blobs clutters the variation pattern. Therefore, it is difficult to identify the arm cycles based on the blob width variation.

The heuristics that the peaks of these graphs correspond to the point in a swimming cycle when the swimmer stretches furthest does not prove to be effective. Table 1 shows that the number of cycles resulting from the experiment is higher than the actual number. As a result, the experimental cycle lengths are shorter than the actual ones. Furthermore, the window size and shifting distance have to be adjusted to achieve more accurate cycle length estimation for each swimming style. In other words, there are no single set of parameter values that are effective for all swimming styles.

5 Discussion

Binary foreground blobs do not provide sufficient information to analyze swimmers' motion. Unlike the background condition used in the human gait analysis experiments in [6] [5] [7], the swimming pool videos have a much noisier background with high frequency moving elements such as water ripples and splashes. As a result, the background subtraction algorithm cannot extract the swimmer's blob accurately. Moreover, binary blobs lack the details that support the recognition of different parts of the human body. Liao et al. [10] has proposed a color-based segmentation approach for swimming style classification. We could use this method to obtain more noise-free foreground images.

The above method could achieve a higher level of accuracy in calculating stroke cycles if we observe all the variations of blob areas, widths and heights. The vertical variation of the swimmer's blob dimension could provide cues to estimate the phase of the arm's cyclic motion. As in [6], Collins chooses either the width or height variation, whichever having a higher amplitude.

A method to estimate more accurately the phase of the arm's cyclic motion is to track the angle between the arm and the body. BenAbdelkader et al. [11] proposes a method of calculating the stride

and height of a person using a formula relating these parameters to the dimension of their silhouette image. Similarly, if the arm length, the upper body length, the angle between these two parts, and the dimension of the silhouette are related by a mathematical model, the phase of the cyclic motion could be calculated.

Currently, the speed of swimmers is converted from pixels per frame to a conventional velocity unit (meters per second) using a known ratio of pixel to distance unit and the frame rate. Having to know the ratio of pixel to distance unit is inconvenient, as this ratio changes with the distance of the swimmer to the camera. A further improvement to solve this tracking problem is to identify the lane markers distributed uniformly along each lane. Usually the distance (in meters) between successive markers are known. Once the markers are identified, we can track the speed of swimmers relative to the markers.

6 Conclusion and future work

We have presented a feasibility study on applying cyclic human gait analysis techniques to tracking swimmer's in noisy background with rippling water. The experimental results show that the area and dimension of segmented binary foreground blobs can be used to track swimmers' positions, but not sufficient for detecting cyclic motion, such as arm movement.

Future work can complement or replace our current approach. An approach is to rely on a mathematical relationship between the arm length and the upper body length to detect the phase of motion. In addition, a color-based foreground segmentation approach is an alternative to segmenting the upper body of swimmers from a different water color in the background.

References

- [1] T. B. Moeslund and E. Granum, "A Survey of Computer Vision-Based Human Motion Capture," *Computer Vision and Image Understanding: CVIU*, vol. 81, no. 3, pp. 231–268, 2001.
- [2] P. Tsai and M. Shah and K. Keiter and T. Kasparis, "Cyclic motion detection for motion based recognition," *Pattern Recognition*, no. 12, pp. 1591–1603, 1994.
- [3] D. Ormoneit, M. Black, T. Hastie and H. Kjellstrom, "Representing cyclic human motion using functional analysis," *Image and Vision Computing*, vol. 23, pp. 1264–1276, Dec 2005.

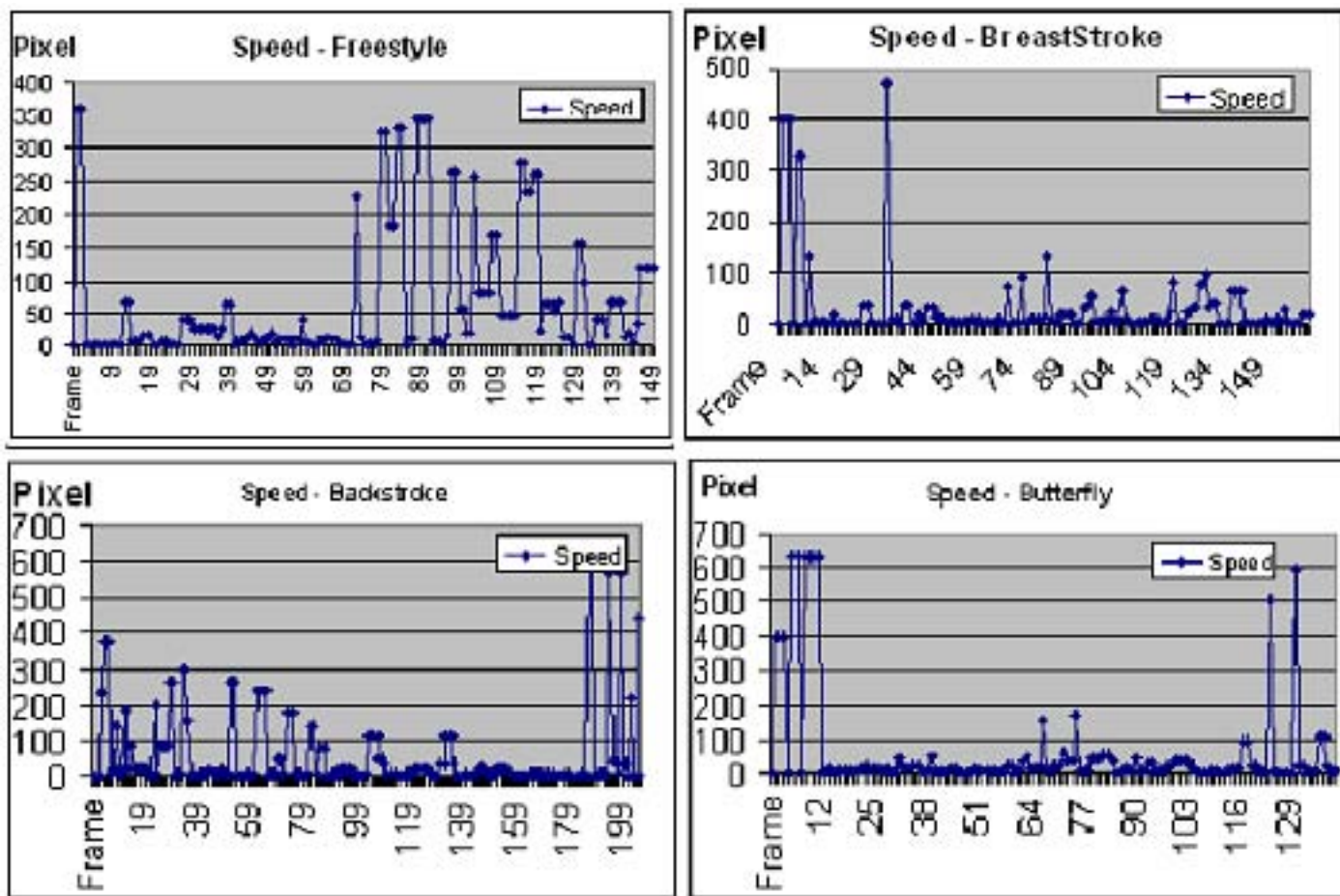


Figure 3: Speed calculated for different swimming styles. The swimming speeds of the breaststroke and butterfly are more stable than the other two.

- [4] R. Urtasun, D. Fleet and P. Fua, "Monocular 3-D Tracking of the Golf Swing," *Conference on Computer Vision and Pattern Recognition, San Diego, CA*, vol. 1, pp. 932–939, June 2005.
- [5] A. Kale, A. Sundaresan, A. Rajagopalan, N. Cuntoor, A. RoyChowdhury, V. Kruger, and R. Chellappa, "Identification of humans using gait," *IEEE Transactions on Image Processing*, vol. 13, Issue 9, pp. 1163 – 1173, September 2004.
- [6] R. Collins and R. Gross and J. Shi, "Silhouette-based human identification from body shape and gait," *Proceedings of IEEE Conference on Face and Gesture Recognition*, vol. 00, p. 366, 2002.
- [7] L. Wang, T. Tan, H. Ning and W. Hu, "Silhouette Analysis-Based Gait Recognition for Human Identification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 12, Dec. 2003, vol. 25, pp. 1505–1518, 2003.
- [8] P. KaewTraKulPong and R. Bowden, "An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection," in *2nd European Workshop on Advanced Video-based Surveillance Systems. Kingston upon Thames*, 2001.
- [9] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *CVPR*, pp. 2246–2252, 1999.
- [10] W. Liao and Z. Liao and M. Liu, "Swimming Style Classification from Video Sequences," *16th IPPR Conference on Computer Vision, Graphics and Image Processing (CVGIP 2003), Kinmen, R.O.C.*, Aug. 17-19 2003.
- [11] C. BenAbdelkader, R. Cutler and L.S. Davis, "Person Identification Using Automatic Height and Stride Estimation," in *ICPR*, vol. 4, pp. 377–380, 2002.

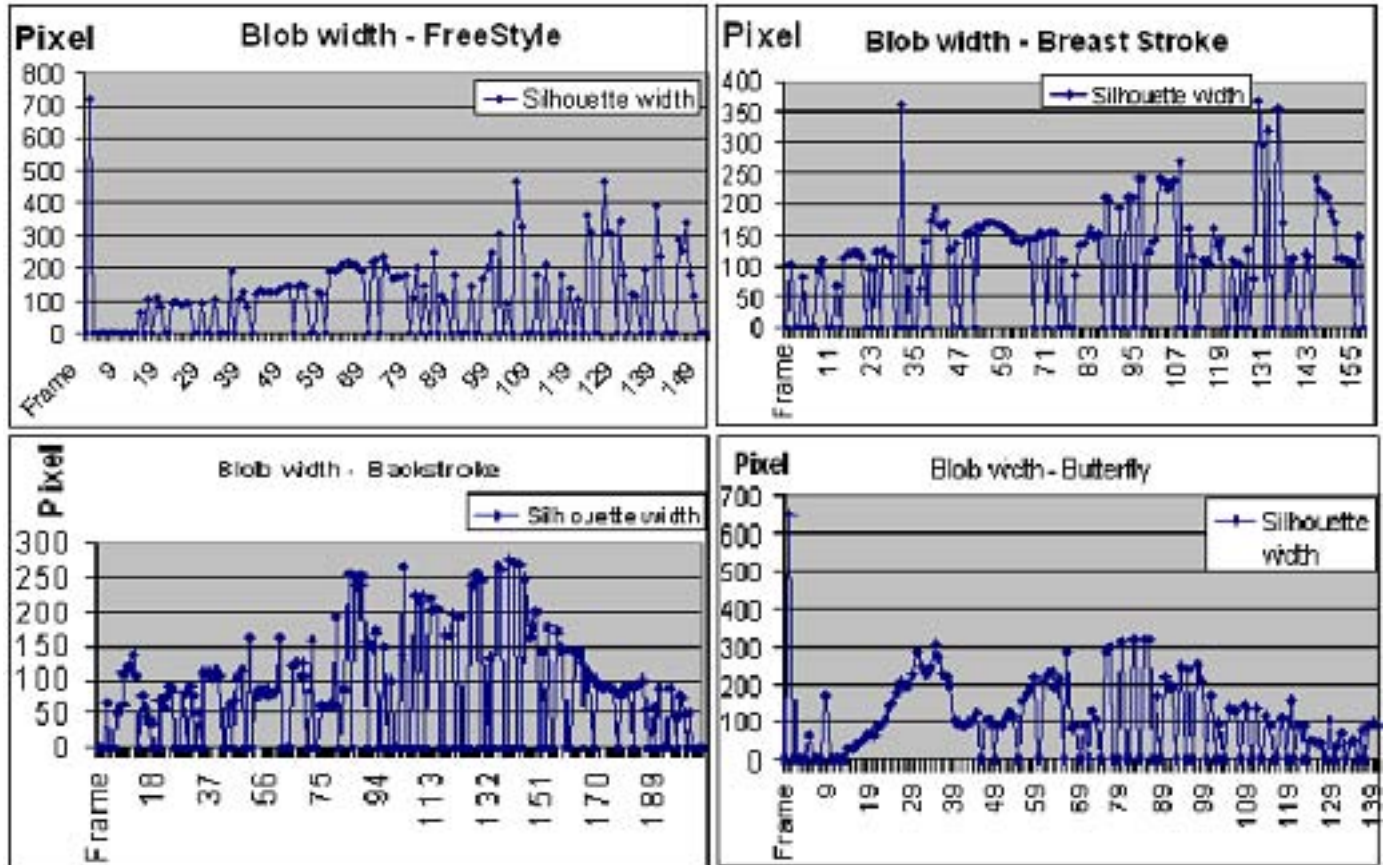


Figure 4: Blob widths calculated for different swimming styles. Noise blobs clutter the periodical variation of swimmer blob widths.

Style	Experimental	Actual
Freestyle	15, 22, 25, 28, 31, 37, 40 ... 142, 145, 148	32, 61, 85, 112, 134
Breaststroke	27, 40, 56, 73, 84, 92, 98, 104, 110, 116, 122	13, 56, 95, 139
Backstroke	10, 15, 17, 19, 21, 23, 25, ... 201, 203, 205	14, 54, 93, 132, 180
Butterfly	31, 63, 85, 105, 129	34, 69, 108

Table 1: The experimental end-of-cycle frame numbers are largely different from the actual ones across various swimming styles.

Accounting for User Familiarity in User Interfaces

C. A. D'H Gough, R. Green, M. Billinghurst

HITLabNZ, University of Canterbury, New Zealand

Email: Christiaan.gough@hitlabnz.org

Abstract

Previous work discussed a model of cognitive distance with the novel concepts of “tech bias”, “velocity” and “inertia”. This paper examines the human aspects of the model by seeking to verify the expected user familiarity behaviour.

It describes a pilot study that suggests the model presented allows for a very high degree of confidence in predicting the effect a user’s familiarity with a problem domain and specific implementation will have on their perception of the directness of the user interface, allowing for greater insight into the construction of optimally effective novel Augmented Reality interfaces.

Keywords: LaTeX style, conference paper, New Zealand conference

1 Introduction

Direct Manipulation is an approach to designing user interfaces, which forms the basis of Graphical User Interfaces.

A good understanding of how Direct Manipulation works is essential in engineering optimal user interfaces; especially in cases such as Augmented Reality, where the interfaces are often novel and highly unusual. A model allowing the prediction of the effectiveness of such novel interfaces prior to construction may save much time in their construction and subjective evaluation.

A previous paper¹ presented a model of the relationship between the user and computer in a Direct Manipulation interface. This model related cognitive distance with user familiarity and the novel concepts of “tech bias”, “velocity” and “inertia”. This model may be used to compare user interfaces and explain or predict differences in the degree of “directness” or “distance” perceived by the user. The model defines the difference between perceived distance and directness as being “User Factors” – primarily that of “User Familiarity”.

In this paper these “Human Factors” are examined more closely, and studies intended to verify the exact effect of these factors are discussed.

2 The model

The model describes 2 key indices – the index of directness and the index of distance. The difference between the index of distance and directness is that of the human factors that contribute to a user’s perception of how direct a user interface is.

2.1 Index of Distance

The first of these indexes is the Index of Distance (S), which may be used to predict the distance a proposed user interface may present.



Cognitive Distance³ is a measure of the gulfs of execution and evaluation – the conceptual gap between the user’s ideas and intentions, and the way in which they are expressed to, or represented by, the system.

Tech bias ($T:(0 < T < 1)$) is defined as “a measure of how well a given device succeeds in the role for which it is intended”¹.

The index of distance scales the cognitive distance of the input and output channels by the Tech Bias of those channels respectively, as shown in equation (1):

$$S = \left(\frac{S_{is} + S_{ia}}{T_i} \right) + \left(\frac{S_{os} + S_{oa}}{T_o} \right) \quad (1)$$

Mature technologies - such as CRT and LCD displays, mice and keyboards are effective at providing their intended experience and, as such, tend to have a high tech bias. Conversely, less commonplace technologies usually have a relatively low tech bias.

In most cases the primary aim of developing an interface is to minimise distance irrespective of user experience, due to the variability of a large user base.

The index of distance is therefore useful for comparing or considering user interfaces in terms that ignore the user factors, such as in the case of engineering a UI for mass-market acceptance.

2.2 Index of Directness

The sensation, as perceived by a user, of increased usability and interactivity provided by a good DM user interface is known as “directness”.

The components of directness are those of the cognitive “distance” between the user and the computer (S), and certain user-related factors (U).

The Index of Directness (D) describes how direct a given user perceives a given implementation of a given user interface to be. It is computed by scaling the index of distance by user factors (U : ($0 < U < 1$)):

$$D = U \times S \quad (2)$$

These user factors (U) were previously defined as familiarity with the user interface (F : ($0 < F < 1$)), yielding the following complete model:

$$D = \frac{1}{F} \times \left(\left(\frac{S_{is} + S_{ia}}{T_i} \right) + \left(\frac{S_{os} + S_{oa}}{T_o} \right) \right) \quad (3)$$

The Index of Directness is an important measure when dealing with a specific, specialised user scenario; where the overall perceived directness might be more relevant than the cognitive distance alone.

2.3 Application

The minimum attainable distance of a given UI is determined by the semantic and articulatory components of cognitive distance of the input and output channels, and the degree to which it is possible to achieve this theoretical minimum is governed by the user factors and tech bias of the hardware used.

This paper uses two layers of interaction – semantic and articulatory, but other common configurations could be used^{5,6,7}.

Due to the inherent difficulties of deriving meaningful values for any of the coefficients used in the model, any evaluation of indices using this model should be used relatively rather than absolutely.

For example, it should not be assumed that an index of directness computed for one case may necessarily be compared directly with another, unless care were taken to use the same scales, assumptions and methodology in both cases.

2.4 Velocity of Mixed Interfaces

An interesting observation may be made in the case of applications where the user is exposed to “mixed distance interfaces”, where various elements of the interface have differing distances.

A good example is that of a recording studio application, where a part is implemented tangibly as a “mixing desk” and a part is implemented via a traditional GUI, mouse and keyboard.

Such mixed-distance interfaces are a sensible approach to improving directness, as they allow a commonly used subset of tasks or operations to have a lessened cognitive distance without sacrificing the flexibility of a more traditional user interface for the less common tasks.

In such cases, it is useful to consider the change of distance that the user must overcome when switching focus between the interface elements. Such variations in distance within an interface can be described as “velocity”.

By taking a weighted average of the Index of Distance for each of the interface types, we can derive a single overall Index of Distance and Index of Directness for the whole interface. This in turn means the theoretically optimal “blend” of interface types can be determined using linear programming.



Figure 1: A typical recording studio application represents a good mixed-distance user interface.

2.5 Inertia

If a user interface is significantly altered in order to improve distance, it must be determined if the gains in directness due to decreased distance are greater than the loss of directness caused by the decreased user familiarity. A small improvement in the distance of a system used by very expert users may not be enough to counter the expertise lost in changing the interface, resulting in a net loss of perceived directness to the user.

Thus, any reductions of distance in an existing user interface must be large enough to overcome the “inertia” of the users’ experience if it is to be a worthwhile improvement without requiring re-learning by the users.

For example, air traffic controllers spend a long time attaining expertise in using their systems. Because these systems are complex and because the safety of

hundreds of lives relies on their effective use, there is much research on improving the user interfaces in order to reduce distance. It would be possible to engineer a new interface that greatly reduced distance using the Index of Distance; but in doing so, much of the acquired directness of the system by the controller may be lost.

In this case the index of directness should be used instead, in order to assess the improvements in light of the inertia of the controller using the system.

It is possible to argue that the primary focus should always be that of directness, as new systems may be re-learned and thus, with time, a new expertise may be joined with the decreased distance to achieve the most optimal possible usability. But consider that in some cases the user may have so much inertia that it is almost impossible to overcome.

For example, surgeons are provided important information via auditory cues during an operation, such as heart rate. Surgeons become so expert at using this system that their use of the interface is almost completely subconscious.

If the interface were re-engineered in such a way that this information was no longer provided, it could result in life-threatening performance decreases for the surgeon that are unable to be re-learned. Any replacement would in essence be a substitute, rather than a replacement, for the auditory approach.

2.6 User Factors

Previous work suggested that the user's sense of directness will be inversely proportional to their level of experience^{2, 3} with the system because, as users become familiar with the interface, less cognitive effort is required to express their desires³.

The user factors that differentiate the index of distance from the index of directness were therefore previously expressed¹ as the reciprocal of user familiarity:

$$U = \frac{1}{F} \quad (4)$$

3 New Model

This paper proposes an expanded definition of the user factors. It was reasoned that the user's familiarity with the problem domain of the application would be equal in effect to that of familiarity of the implementation of the application used – all other factors held constant - when determining the user's perceived directness with a given application.

The value of U was therefore updated to take the following form:

$$U = \frac{2}{F_d + F_i} \quad (5)$$

where F_d represents the familiarity of the user with the problem domain, and F_i represents the familiarity of the user with the specific application in question.

4 Pilot study

A pilot study was performed to gain insight into the validity of this model. Participants were provided with a URL of a website containing a questionnaire.

Participants were able to log in to this website and answer a series of questions regarding their degree of experience with, and perception of, various implementations of operating systems and file operation environments. Several of these questions gave insight into the participant's experience with 4 specific operating systems – Windows XP, Mac OSX, Linux and Command Line Interfaces such as DOS.

Table 1: Questions asked in the pilot study. These questions were presented to the participant 4 times with a different OS replacing "X".

1	How familiar are you with X? [1 = very unfamiliar, 5 = very familiar]
2	How would you rate your mastery of X? [1 = not good, 5 = very good]
3	How competent do you feel in performing tasks with X? [1 = very incompetent, 5 = very competent]
4	How much do you enjoy performing tasks with X? [1 = not very much, 5 = very much]
5	How confident are you when using X to perform tasks? [1 = very unconfident, 5 = very confident]
6	If you had to give an overall rating of X, what would it be? [1 = very bad, 5 = very good]
7	How easy do you feel it was to learn to use X? [1 = very difficult, 5 = very easy]
8	How easy do you feel it is to learn new features of X? [1 = very difficult, 5 = very easy]
9	How confident are you in your ability to retain your current mastery of X? [1 = very unconfident, 5 = very confident]
10	How eager would you be to demonstrate the use of X or train novices in using X? [1 = not very eager, 5 = very eager]
11	How much do you want to explore the more powerful aspects of X? [1 = not at all, 5 = very much]
12	How easily do you feel you can achieve a given task using X? [1 = not very easily, 5 = very easily]
13	How much do you feel that X is a tool or extension of yourself, rather than part of the task to be achieved? [1 = not at all, 5 = very much]

The remaining questions were intended to gather an appreciation of the participant's perceived directness of the operating systems, based on the list of proposed benefits of a good DM interface described by Shneiderman. All questions were to be answered using a Likert Scale of 1-5.

The questions were duplicated exactly for each operating system so that, in effect, each participant was completing the same questionnaire 4 times for different Operating Systems.

5 Results

The results from 22 participants were processed in such a way that 88 samples were obtained, where each sample represented a set of results of one participant's rating of their experience and perceived directness of an individual Operating System. Each of these results are represented on figures 2, 3 and 4 as a single diamond.

The results of the questions pertaining to the participant's familiarity with a given OS were averaged for each sample to obtain a value for their F_i for that OS, and the remaining questions of that sample were averaged to represent the participant's perceived Directness (D) for that sample.

The resulting correlation between F_i and measured D gave a good correlation ($R=0.863$, $R^2=0.745$) (fig 2)

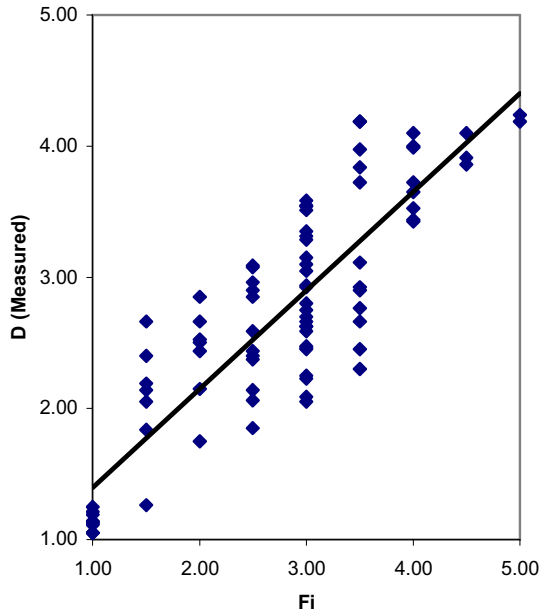


Figure 2: F_i versus Measured D

The F_d of each participant was then computed by averaging the F_i of each of the 4 OS samples for that participant. Plotting the correlation between each

participant's F_d and the D for each of their samples gave a low correlation ($R=0.448$, $R^2=0.201$)(fig. 3).

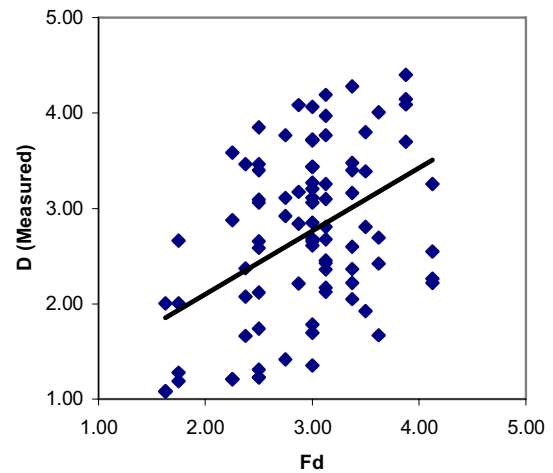


Figure 3: F_d versus Measured D

Finally, the U for each sample of each participant was computed using the method proposed by this paper – by averaging the F_d and F_i for each sample.

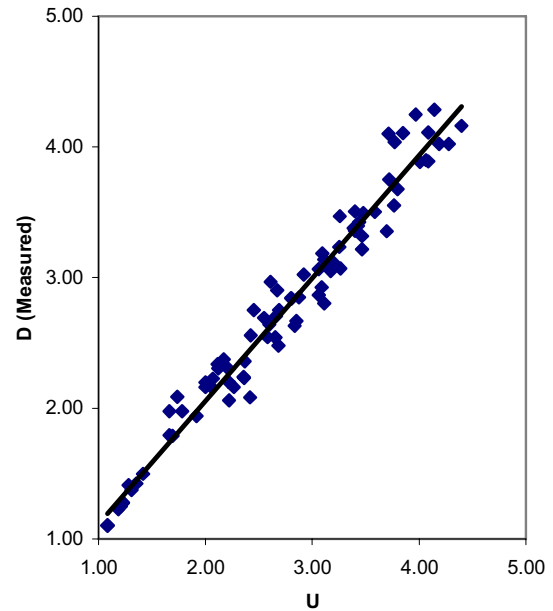


Figure 4: U versus Measured D

The resulting U for each sample of each participant was very highly correlated ($R=0.983$, $R^2=0.967$)(fig. 4).

6 Conclusion

The results of this pilot study suggest that a given user's perception of directness of a given user interface may be accurately predicted using the model described in this paper.

This model is significantly more accurate than the traditional approach of simply assuming perceived directness will be proportional to their familiarity with the interface alone.

The pilot study suggests that the model described may allow confidence of over 90%, although this needs to be verified with more rigorous experimentation.

A good understanding of how this effect works is essential in engineering optimal user interfaces; especially in cases such as Augmented Reality, where the interfaces are often novel and highly unusual. A model allowing the prediction of the effectiveness of such novel interfaces prior to construction may save much time in their construction and subjective evaluation.

7 Future Work

This study was an initial stage in verifying and examining the theories described by Gough et al.

More exhaustive studies are currently being carried out to provide greater insight into the results indicated by the pilot study described in this paper. These experiments are as follows:

7.1 Experiment 1: File operations

This experiment is to be a more exhaustive and rigorous version of the pilot study described in this paper.

The problem domain will be restricted to that of file operations alone, rather than more general usage of Operating Systems.

Participants will be asked to perform a variety of file operation tasks using a variety of approaches, including some approaches that will be custom-implemented so as to allow greater focus on the relationship between Fd and Fi.

The purpose of this experiment is to gain insight into the role of user familiarity with a problem domain and implementation, and to replicate the results of this paper with more accuracy and rigour.

7.2 Experiment 2: Creative content

Participants will be asked to "mix" a song based on content provided to them.

Participants will have varying experience in the use of computers, audio editing and mixing, and in performing and creating music.

Mixing will take place on a mixing desk alone, a computer alone, and a mixed-distance interface consisting of an automated mixing desk coupled with an interoperable software environment on a connected computer.

Once again, the purpose of the experiment is to gain insight into the role of user familiarity with a problem domain and implementation, and to replicate the results of this paper with more accuracy and rigour.

There will, however, be additional scope to gain insight into the effect of the effect of mixed-distance user interfaces on perceived Directness, and specifically the interrelation between the indices of distance and directness under a mixed-distance situation.

7.3 Experiment 3: DB Query

Participants will be asked to perform a series database search queries. The queries will be via traditional user interfaces such as a web-based search engine, an SQL command string and a form-based Access GUI.

Participants will also be provided with several new graphical and tangible approaches based on both new existing metaphors.

The benefit of this research will once again be primarily that of verification of the existing model, but will also allow unique insight into other potential factors unaccounted for at present, as well as into the interplay of the indices of directness and distance.

7.4 Experiment 4: IDE Usage

The final experiment will require users to perform a series of common tasks using development environments. The questions listed in table (1) will be asked of the participants, and correlated in the same way as the pilot study and previous experiments.

8 References

- [1] Gough, C. Green, R. Billinghurst, M., "Better Realising Direct Manipulation", Image and Vision Computing New Zealand 2005, pp. 455-460.
- [2] Shneiderman, B., "The Future of Interactive Systems and the Emergence of Direct Manipulation", Behaviour and Information Technology 1982, v.1 n.3, pp. 237-256
- [3] Hutchins, E. Hollan, J. Norman, D., "Direct Manipulation Interfaces", Human-Computer Interaction, Volume 1, 1985. pp. 311-338
- [4] Frohlich, D., "The history and future of direct manipulation" Behaviour & Information Technology 12, 6, 1993. pp. 315-329.
- [5] Nielsen, J., "A Layered Interaction Analysis of Direct Manipulation", 1992

- [6] Hix, D. and Hartson, H., "Developing User Interfaces". John Wiley & Sons, Inc, 1993
- [7] Taylor, M., "Layered protocol for computer-human dialogue", I: Principles. International Journal of Man-Machine Studies, 28, 1988. pp. 175-218
- [8] Antifakos, S. "Improving Interaction with Context-Aware Systems", Selected Readings in Vision and Graphics, volume 35, 2005
- [9] Ishii, H., Kobayashi, M., Grudin, J. "Integration of Interpersonal Space and Shared Workspace: ClearBoard Design and Experiments," ACM Transactions on Information Systems (TOIS), ACM, Vol. 11, 1993
- [10] Nelles, C. "Graphical vs Tangible User Interface", unpublished PhD thesis, eingereicht am Fachhochschul-Diplomstudiengang, Medientechnik und Design, Hagenburg, 2005

Image Denoising Using a New Line-Field

Ngoc-Thuy Le, Kah-Bin Lim

Control & Mechatronics Lab , Dept. of Mechanical Eng., National University of Singapore.

Email: lnthuy@nus.edu.sg

Abstract

In this paper, we propose an iterative denoising algorithm using a line-field based filter with the Simulated Annealing scheme. Different from the line field introduced by Geman and Geman [1], our modified line field corresponds to a limited interval of the intensity difference between a pixel and its neighbors. By applying the maximum a posteriori approach, an adaptive filter is constructed based on the modified line field to remove the noise while preserving the edge of image. The convergence of the modified line field and the annealing schedule guarantee the convergence of the algorithm. The proposed algorithm is efficient when compared with many existing methods such as VisuShrink [2], SureShrink [3], and BayesShrink [4].

Keywords: Markov random field (MRF), simulated annealing scheme, line field, maximum a posteriori.

1 Introduction

In the field of image denoising, there are two main approaches: processing the image in the spatial domain or in the time-frequency domain using the wavelet transform.

The wavelet transform is appreciated for denoising images because the white noise in the signal is still the white noise in the transform domain while it will concentrate into few coefficients in the wavelet domain [5]. This important principle which is capable for separating the signal from noise makes wavelet transform be appropriate for estimating data with sharp discontinuities such as the edge of images. The efficiency of the approach depends on choosing a proper shrinkage threshold. There were many efforts to estimate the shrinkage threshold such as: RiskShrink [2] using a soft-threshold operator and minimizing the mean squared error; VisuShrink [2] applying a global optimal threshold in the mini-max sense of RiskShrink; SureShrink [3] minimizing Stein's unbiased risk estimate; or BayesShrink [4] performing a data-driven, subband-dependent threshold. However, this approach costs much computational time for wavelet and wavelet inverse transformation.

The approach in the spatial domain might be more competent in term of computational time. Furthermore, its algorithms are easy to be generalized and combined with the other image processes such as: segmentation, pattern recognition, or deblurring. The algorithms in the spatial domain are based on the idea of locally smoothing the image with different smooth coefficients. A classical method is

using a median filter to suppress the noise but it also blurs edges and details of image. Many considerable works have overcome this effect by switching among several median-filters based on some criteria [6], [7], and [8]. Recently, Katkovnik [9] has proposed an efficient denoising method using the local polynomial approximation (LPA) with the adaptive window size estimated by the intersection of confidence intervals (ICI) rule. However, these works often estimated the image by the average of various smoothing directions that could blur the image at its edges [6], [9].

Inspiration from the similarity about the locally dependent characteristic of the image and the Markov chain, Besag [10] has proposed a valid probability structure to model mathematically the image. By adding a new process, called the line process, to this model, Geman and Geman [1] has made the model more powerful in removing the noise while preserving the detail of the image since the line process has driven the smoothing process appropriately. Applying different iterative schemes, such as Simulated Annealing (SA) scheme [11] or Iterated Conditional Modes (ICM) scheme [12], to these models has resulted in the efficient denoising algorithms that have had a better capability for preserving the detail of the image. However, because of the convergence condition, these algorithms required hundreds of iterations. It will result in a considerable computational time. By using a variant line field instead of the original line field, we could distinguish a pixel at the edge of image from the noise. Therefore, it is capable for accelerating the convergence speed and reduce the computational

time significantly. The fact that the stop criterion is reached after about ten iterations makes our algorithm less able to blur the image while removing effectively the noise.

This paper is organized as follows. Section II describes briefly the image model based on the MRF model and suggests a modified line field which distinguishes a point at the edge from the noise efficiently. Section III proposes an iterative denoising algorithm using the SA scheme. Some experimental results and its comparison with the other works could be found in this part. Section IV concludes the paper.

2 Theoretical Developments

2.1 MRF and Image Modeling

We model here our problem as the following form:

$$g = f + n \quad (1)$$

In equation (1), g is the observed image, f is the noisy-free image, and $n \sim N(0, \sigma_n^2)$ is an additive white Gaussian noise. Hence, the conditional probability of the observed image given the original image is correspondent to a Gaussian distribution (equation (2)):

$$P(g_i|f_i) = \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp \left\{ -\frac{1}{2\sigma_n^2} [g_i - f_i]^2 \right\} \quad (2)$$

In this model, the image is regarded as a pair of 2-D Markov random fields, the intensity field F and the line field L , for the meaning that each element only depends on its neighbors. While F is the real field representing the intensity at each pixel, L is the imaginary field representing the bond between pixels. By adding the line field to the image model, the edges of the image are distinguished from the noise thank to the number of bonds that a pixel has with its neighbors. Therefore, the denoising process does not oversmooth the image while removing the noise effectively. Besag (1974) [10] has proposed a valid probability structure of the intensity process F in order to model an image. The probability has the form:

$$\begin{aligned} P(f_i|f_j : j \neq i) &= P(0) \exp \left\{ \sum_{1 \leq i \leq n} f_i G_i(f_i) \right. \\ &+ \sum_{1 \leq i \leq n} \sum_{i \leq j \leq n} f_i f_j G_{ij}(f_i, f_j) \\ &\left. + f_i \dots f_n G_{i\dots n}(f_i, \dots, f_n) \right\} \quad (3) \end{aligned}$$

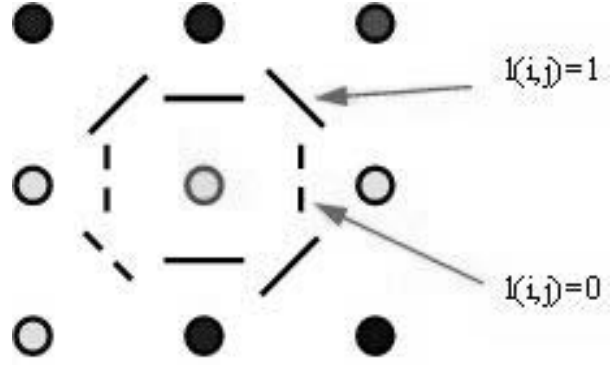


Figure 1: Line-field model: the neighbors of a pixel and the bonds between them ($l(i, j) = 1$ if existing the bond; otherwise $l(i, j) = 0$).

From this general form, we can construct specific models. Regarding a simple case in which the first order terms of G are linear, the second order terms of G are constant, and the others are equal to zero, we obtain an auto-normal model [13]:

$$\begin{aligned} P(f_i|f_j : j \neq i) &= \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp \left\{ -\frac{1}{2\sigma_i^2} \left(f_i - \sum_{j \neq i} \beta_{ij} f_j \right)^2 \right\} \quad (4) \end{aligned}$$

where β_{ij} is equal to zero unless j is a neighbor of i and there is a bond between them. The conditional variance σ_i^2 characterizes for the local smoothness at the pixel i and should be increased at the edges of the image. These parameters are determined by bonds around the pixel, generally speaking, by the line field. The line field of an image is an imaginary field which is constructed from the intensity field of that image. If there is "no difference" between the intensity of a pixel and that of its neighbor, it is said that it does not exist a bond between them ($l(i, j) = 0$); and otherwise $l(i, j) = 1$ (as shown in figure 1). Then, the line field is a binary random field.

2.2 MAP Approach

The problem can be solved with the maximum a posteriori (MAP) approach by applying the annealing schedule into (4). The annealing schedule implies an iterative algorithm controlled by a "temperature" parameter T which decreases slowly with respect to the iteration step k . The conditional probability of the intensity at a pixel given that at the others can be modified as:

$$P(f_i|f_j : j \neq i)$$

$$= \left(\frac{1}{\sqrt{2\pi\sigma_i^2}} \exp \left\{ -\frac{1}{2\sigma_i^2} \left[f_i - \sum_{j \neq i} \beta_{ij} f_j \right]^2 \right\} \right)^{\frac{1}{T(k)}}$$

It is proved that the temperature $T(k)$ should be satisfied the bound:

$$T(k) \geq \frac{c}{\log(k+1)} \quad (5)$$

Adding the parameter $T(k)$ is to guarantee the convergence of the iterative algorithm. The constant c is independent to the step k and is capable for controlling the speed of convergence. It is necessary to choose an appropriate value of c to achieve the desired precision while requiring as less effort of computation as possible. Following the MAP approach, we apply the Bayesian formula:

$$P(f_i|g_i, f_j : j \neq i) = P(g_i|f)P(f_i|f_j : j \neq i)$$

or

$$\begin{aligned} -\log P(f_i|g_i, f_j : j \neq i) &= \text{const} \\ + \frac{1}{2\sigma_i^2 T(k)} [f_i - \sum_{j \neq i} \beta_{ij} f_j]^2 + \frac{1}{2\sigma_n^2} \end{aligned} \quad (6)$$

Then, the problem becomes an easier one to solve.

2.3 Modified Line Field

In the model suggested by Geman and Geman [1], the probability of the existence of a line between two pixels ($l(i, j) = 1$) is an invariant distribution which covers the whole variation interval of intensity difference. It may lead to the confusion between a noise and a point at the edge. In our model, we call a pixel is a noise if its intensity differs from those of its neighbors at the same level while the intensity of a point at the edge might differ from those of its neighbors at various levels. Thus, by modifying the distribution which covers a limited interval, a point at the edge could be distinguished with the noise. At each iteration step k , the distribution covers at a different interval and the summation of them must cover the whole variation interval of intensity difference. To guarantee the convergence of the algorithm, the variance of the distribution $\sigma_\Delta^2(k)$ should decrease following the decrease of its mean $\mu_\Delta(k)$. The probability of the line is given in equation (7):

$$\begin{aligned} P(l(i, j) = 1|f, l(m, n) : (m, n) \neq (i, j)) \\ \sim \exp \left(\frac{[|f_i - f_j| - \mu_\Delta(k)]^2}{2\sigma_\Delta^2(k)} \right) \end{aligned} \quad (7)$$



Figure 2: The noise-free Lena image (top-left), the noisy image (top-right) $\sigma_n = 20$ ($PSNR = 22.14dB$), and the results of denoising processes using (6) with the original (bottom-right) ($PSNR = 29.70dB$) and modified (bottom-left) ($PSNR = 30.77dB$) line field.

Once the line field is determined, the parameters β_{ij} and σ_i^2 could be calculated from the line field. As stated above, β_{ij} different to zero implies a bond between pixel i and pixel j ($l(i, j) = 1$):

$$\beta_{ij} = \frac{l(i, j)}{\sum_{j \neq i} l(i, j)} \quad (8)$$

We also know that σ_i^2 increases when scanning toward the edges of the image. We construct here a new coefficient to distinguish a point at the edge and a noise:

$$\alpha_i = \exp \left[\frac{(\sum_{j \neq i} l(i, j) - N)^2}{2\sigma_l^2} \right]$$

where N is the number of neighbors around the pixel i and σ_l^2 is the variance of the distribution of the number of lines around a pixel. α_i is called the noise coefficient. It is high if the noise exists at the pixel i and low if otherwise. Hence, σ_i^2 varies inversely with α_i . For instance, σ_i^2 might be chosen simply:

$$\sigma_i^2 = \sigma_n^2 (1 - \alpha_i) \quad (9)$$

Table 1: PSNR[dB] Results of VisuShrink, SureShrink, BayesShrink, equation (6) with Geman’s line field and the proposed algorithm.

	$\sigma_n = 10$	$\sigma_n = 20$	$\sigma_n = 30$
Noisy Image	28.18	22.14	18.62
VisuShrink	28.76	26.46	25.14
SureShrink	33.28	30.22	28.38
BayesShrink	33.32	30.17	28.48
Equation (6)	31.78	29.70	28.12
Our algorithm	34.18	30.77	28.95

3 Experiments and Results

Following the theoretical developments above, an iterative algorithm is proposed to solve the denoising problem:

- Step 1: Set $k := 1$;
- Step 2: Define the temperature $T(k)$, the variance $\sigma_{\Delta}^2(k)$ and the mean $\mu_{\Delta}(k)$ of the modified line field distribution;
- Step 3: Calculate the binary line field $l(i, j)$ following (7);
- Step 4: Determine the parameters β_{ij} and σ_i^2 following (8), (9);
- Step 5: Estimate the intensity at each pixel from (6) by applying the MAP approach;
- Step 6: Set $k := k + 1$ and go to step 2 if the stop criterion is not satisfied.

For simplicity, we can choose the stop criterion being the number of iteration steps. In the other word, the algorithm is finished after a specified loop number N_{loop} . The accuracy and effectiveness of the algorithm is resulted from choosing properly the parameters $T(k)$, $\sigma_{\Delta}^2(k)$, and $\mu_{\Delta}(k)$. In our experiments, we fix the decrement of $\mu_{\Delta}(k)$ and $\sigma_{\Delta}^2(k)$ at k^{-2} and change $T(k)$ according to the noise variance σ_n^2 . From experiment, we found the relationship between the noise variance and the constant c of the "temperature" $T(k)$ to optimize the algorithm. The theoretical study about this relationship could be an interesting prospective work.

To compare effectively with the existing methods, the proposed algorithm is applied in the famous Lena image corrupted by additive white noises with different variances (as shown in figure 2). The image size is 512x512. The experimental results were compared with those of the other methods in term of Peak Signal Noise Ratio (PSNR). The quantitative performance comparison in Table 1 shows that our method is highly competent

with denoising techniques in the literature such as VisuShrink, SureShrink, BayesShrink. Moreover, our proposed algorithm, which is realized without steps related to the wavelet and wavelet inverse transformation, requires less efforts on computation. In addition, the modified line field, whose distribution has been changed at each iteration, has increased the convergence speed and reduced the computational time significantly. Another advantage of the algorithm is that the stop criterion is reached after about ten iterations. This fact makes our algorithm less able to blur the image while removing effectively the noise.

4 Conclusions

The MRF is an appropriate tool for modeling the image. Adding the line field to the model makes it more powerful in processing the image while preserving image details. The suggested line field helps to distinguish between the noise and the edge of images and results in an efficient denoising algorithm. However, the result might be better by finding a proper scheme for the line field distribution.

To improve the proposed algorithm, we also suggest to combine it with the local polynomial approximation (LPA) driven by the line field so that the denoised image would be even smoother.

References

- [1] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *PAMI*, no. 6.6, pp. 721–741, 1984.
- [2] D. L. Donoho and I. M. Johnstone, "Ideal spatial adaption via wavelet shrinkage," *Biometrika*, no. 81, pp. 425–455, 1994.
- [3] D. L. Donoho and I. M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *Journal of American Statistical Assoc.*, no. 90.432, pp. 1200–1224, 1995.

- [4] B. Y. S. Grace Chang and M. Vattereli, "Adaptive wavelet thresholding for image denoising and compression," *IEEE Trans. Image Processing*, no. 9, pp. 1532–1546, 2000.
- [5] C. Taswell, "The what, how, and why of wavelet shrinkage denoising," Tech. Rep. CT-1998-09, Computational Toolsmiths, Stanford, January 1999. www.toolsmiths.com.
- [6] A. Kondu and J. Zhou, "Combination median filter," *IEEE Trans. Image Processing*, no. 1.3, pp. 422–429, 1992.
- [7] A. B. Hamza and H. Krim, "Image denoising: A nonlinear robust statistical approach," *IEEE Trans. Image Processing*, no. 49.12, pp. 3045–3054, 2001.
- [8] P.-E. Ng and K.-K. Ma, "A switching median filter with boundary discriminative noise detection for extremely corrupted images," *IEEE Trans. Image Processing*, no. 15.6, pp. 1506–1516, 2006.
- [9] K. E. Vladimir Katkovnik and J. Astola, "Adaptive window size image de-noising based on intersection of confidence intervals (ici) rule," *Journal of Math. Imaging and Vision*, no. 16, pp. 223–235, 2002.
- [10] J. E. Besag, "Spatial interaction and the statistical analysis of lattice system," *J. Royal Stat. Soc. Ser. B*, no. B-36, pp. 192–236, 1974.
- [11] F.-C. Jeng and J. W. Woods, "Simulated annealing in compound gaussian random field," *IEEE Trans. Information Theory*, no. 36.1, pp. 94–107, 1990.
- [12] J. Park and L. Kurz, "Image enhancement using the modified icm method," *IEEE Trans. Image Processing*, no. 5.5, pp. 765–771, 1996.
- [13] J. E. Besag, "On the statistical analysis of dirty pictures," *J. Royal Stat. Soc. Ser. B*, no. B-48, pp. 259–302, 1986.

Augmenting Sports Grounds with Advertisement Replacement

D.K. Barrow, R. Green

Department of Computer Science and Software Engineering
University of Canterbury, Private Bag 4800
Christchurch 8140, New Zealand

Email: dba46@student.canterbury.ac.nz

Abstract

Augmented imagery commonly viewed as a subset of augmented reality, is the modification of still images or video through the addition of computer generated virtual objects. Over the last decade augmented imagery has demonstrated its value through commercial, medical and other popular applications such as video games, public television broadcasting and commercial simulations. In this paper we present an augmented imagery application for virtual advertisement which replaces a blank planar region in a given video sequence with commercial advertisement content. The application is built using relatively cheap and in some cases freely available tools which will show the affordability and ease with which such applications can be built. The application uses OpenCV library to capture, manipulate and render in real-time, commercial content within digital video the video stream as well as the accuracy of ARTag tracking libraries to track fiducial markers representing the overlay region. We also address the issue of occlusion using a background-foreground segmentation algorithm to render our overlay as part of the background whenever it coincides with a foreground object. The paper presents details of the requisite concepts and technical implementation which utilises real time video from a USB camera with no specialized hardware support.

Keywords: Augmented Reality, ArTAG, advertisement, Background-Foreground Segmentation, Occlusion

1 Introduction

Augmented reality falls second within the *virtuality continuum* proposed by Milgram and Kishino [12] and deals with the joining of real and virtual objects within a real environment setting such that virtual objects appear to be seamlessly integrated. It has over the past decade grown tremendously in popularity as it lies closer to the real world than the virtual world and as such is being applied across industries in medical, military, and commercial applications particularly in video broadcasting of sports and commercial advertisements. In this paper we will look specifically at augmented imagery, a subset of augmented reality which encompasses the enhancement of digital imagery.



Figure 1: Milgram's Virtuality Continuum.

As illustrated in Figure 2 [6], augmented imagery involves a number of complex processes not limited

to but primarily involving modelling the real environment, determining the relationship between the real and virtual models, calculating the transformations to accurately render virtual objects such that they accurately coincide with real objects and finally the blending of all images into one seamlessly integrated image.

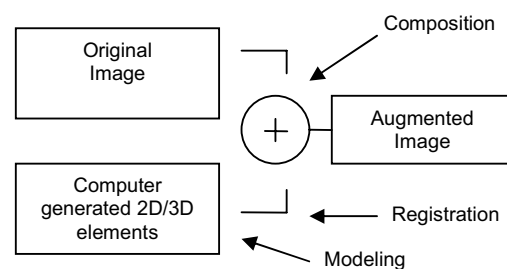


Figure 2: The Basic Augmentation Process.

Our applications seeks to achieve the above at minimal cost requiring no special preparation of the background scene, no excessive camera calibration techniques and employs the use of simple marker based tracking techniques to overcome the complex issues of registration. We also noted in [9] but did not

implement the idea of user invisible marker detection such that content could be rendered more realistically. Below we consider how our application conforms to the basic development model outlined above and then present a more detailed outlay of the application.

1.1 Modeling

Modeling involves expressing both the virtual and real elements of the environment in mathematical terms and storing these expressions as data structures. These descriptions can potentially become quite complex depending on the nature of the augmentation exercise. The modeling process can be broken into two main categories 2D and 3D. In this paper we present a very simple implementation of 2D modeling which will involve storing a rectangular region description of our real environment and destination images, and the virtual overlay images to be rendered. Our virtual advertisement will be described simply as a preformatted image and the destination location as a quadrilateral marked and tracked in the real environment. Realistic 3D augmentation requires description of attributes such as geometry, lighting, surface characteristics and more complex shapes which are outside the scope of this paper.

1.2 Registration

The process of determining the series of transformation required to accurately align virtual and real objects is known as registration and this is essential to producing a realistic and natural effect. Our virtual advertisement replacement program uses 2D to 2D registration which will transform pixel coordinates in the advertisement image to pixel coordinates in our destination image. In order to obtain accurate registration we will use optical tracking through the use of ARTag fiducial markers. We track the centre of the marker to obtain the four vertices where the overlay is to be placed.

1.3 Composition

This refers to the blending of the source and destination images to produce a single seamless image. Each pixel in the final image can be viewed as a combination of the related pixels of the original images. Alpha maps are frequently used to produce a final image that appears real and they function primarily by determining which portions of an overlay image should remain opaque and which portions should be made transparent. Other techniques are used to produce adequate maps and these include static mattes, luma keying, difference mattes and constant colour matting [6]. In this paper we use alpha maps to render the image more realistically.

Composition will also involve the use of background-foreground segmentation to detect those areas of the overlay which occlude foreground objects and to render those pixels into the image background.

2 Applications and Related Work/Current Developments

In this section we look at current and related work being done in the area of augmented reality in general. The Eye-Toy add-on for the Sony Playstation console is a camera which recognizes player's movements and is a first approach to a commercial product. The number of research projects involving augmented reality continues to grow primarily with development environments such as ARtoolkit, ARTag and ARstudio. In [9] Dennis Joele proposes development of a AR system using ARtoolkit, infrared cameras and user invisible markers created by ink which is only visible in the infrared spectrum. VTT Information technology a multimedia group is currently working on several projects including IMMOVE which enables the provision of a new range of intelligent video based services to end users in various mobile/wireless networks including CamBall an augmented virtual table tennis game over Internet/LAN using real rackets and Virtual interiors an interior designing application. In 2003 David Sickinger conducted some work in biomedical field using augmented reality in rendering a virtual skull over a real face.

Augmented imagery applications are being seen from sports to medicine to assisting soldiers out in the field, augmented reality is being applied widely. Popular implementations include first down lines visible during an American football game or superimposed ball trajectory diagrams during cricket broadcasts and virtual studios which have evolved from blue screen compositing techniques replacing virtual sets with computer generated virtual sets which are integrated seamlessly.

An approach very simpler to the one taken here is also presented by Wang, Sengupta, Kumar & Shamar (2005) who utilise Tamasi and Kanade's (1981) method for strong point detection for tracking vs. our chosen fiducial marker system.

3 Design Tools

The two main tools used in the design of this application and the implementation of our approach are OpenCV, a library of programming functions facilitating real time computer vision and ARTag, a fiducial marker system similar to ARtoolkit but with some improvements. In this section we focus on ARTag and the marker detection process and take for granted that you the reader are familiar with OpenCV.

3.1 ARTag

ARTag is a fiducial marker system, a 2D marker and computer vision system. Fiducial marker systems are consist of digitally generated pattern images mounted in an environment and the accompanying algorithm used to detect the patterns. They are used in a

number of systems where the relative position between a camera and an object is required.

ARTag uses digital coding theory to get a very low false positive and inter-marker confusion rate thus allowing it to work with a very small marker size. It also utilises an edge linking method to give greater consistency under poor lighting conditions and to provide occlusion immunity.

Markers patterns are created using a square outline for the exterior and a 36-bit word encoded in the interior of the square all position on a bi-tonal planar surface. To prevent false detection of marker patterns, checksums and forward error correction (FEC) are used to protect a unique identification number contained in each digital word of a pattern providing very low and numerically quantifiable error rates.

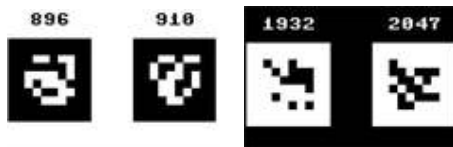


Figure 3: Examples of ARTag markers [2].

ARTag was used in the development of this project because it represents an improvement over its predecessor ARToolkit another marker system. While ARToolkit is very successful and widely used, ARTag represents a significant increase in performance in the area of false positive detection and inter-marker confusion thus decreasing the number of times markers are either confused for one another or falsely detected because of background noise.

ARTag can encode up to 2046 unique IDs without the need to store patterns as is the case with ARToolkit.

3.1.1 ARTag Marker Detection

ARTag uses the four prominent corners of its square border markers to allow the full extraction of the 6 degree of freedoms (DOF) of the relative marker to camera position. ARTag allows both a black border on a white background and a white border on a black background.

Each square border is divided into a 6x6 square grid holding 36 information carrying cells, the whole marker being 10x10 units and having a border thickness of 2 units. Each of the 36 cells are capable of carrying exactly one bit of digital information.

The boundary of the marker is detected by thresholding a greyscale version of the image, performing connectivity of pixels above and below this the obtained threshold and finding those connected objects with a quadrilateral boundary. Once the border has been located the internal region is sampled with a 2D 6x6 array and assigned digital symbols '0' or '1' using the same threshold as above

and to output four 36 bit sequences one for each of the four possible rotation positions. Each 36 bit sequence passes through the FEC stage which can detect and correct some bits.

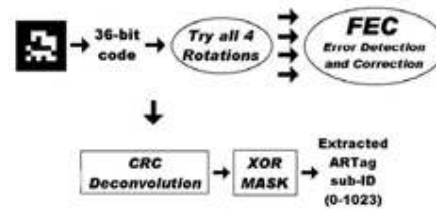


Figure 4: Digital Decoding Process [2].

The result is then analysed by a CRC checksum procedure which verifies its membership within the ARTag marker set. If it does belong to the set then the encapsulated 10-bit ID (sub-ID) number is extracted, combined with the border polarity and a detected marker is reported. This entire process represents the decoding phase.

3.1.2 ARTag Library

In this section we will review some of the more critical and frequently used function within the ARTag library. These functions primarily implement the marker detection and identification process.

In ARTag an object is either a single ARTag marker, or a 2D array. In order to locate markers the following steps are performed:

1. Initialisation

- first call `init_artag()`
- load an array (.cf) file with `load_array_file()` if you want to use arrays
- associate single markers or arrays with objects

2. Frame Processing

search the image for objects using `artag_find_objects()`

for all associated objects, determine whether object can be found in the current frame by calling `artag_is_object_found()`,

if the marker object can be found proceed with relevant action e.g. image rendering

The `init_artag` function is used to initialise the background environment before calling any other functions. `init_artag` takes as argument the width and height of the camera image where the marker is located. The function also take a variable which stores the bytes per pixel (bpp) or colour depth of an image (bpp=1 for greyscale 8-bit, 3=24-bit, 4=32-bit.

```
char init_artag(int width, int height, int bpp);
```

ARTag allows you to associate objects with either single markers or arrays. In order to use arrays, the file containing the array definitions must first be loaded. This is accomplished using the function below.

```
int load_array_file(char *filename);
```

artag_associate is used to associate an array with an object and returns an ID to use in future calls. "frame_name" is the array name as in appears in the .cf file which must be previously loaded. A return value of -1 indicated that the object could not be initialized.

```
int artag_associate_array(char *frame_name);
```

artag_find_objects() is the main call that does all the work. It searches the image passed to it for markers, and then finds those that belong to defined objects. It internally calls the artag_find_marker() function which locates all markers in the image.

```
void artag_find_objects(unsigned char *, char );
```

rgb_greybar is set to 1 for RGB images and 0 for greyscale.

```
char artag_is_object_found(int artag_object_num);
```

This function is used to determine the presence of a given object within an image. It receives a unique object number previously associated with a given marker or array and uses this number to determine whether that marker is present in the image.

```
artag_project_point(int,float,float,float*,float*);
```

The project point function can be used to find what an (X,Y,Z) coordinate in an object maps to in camera image coordinates. The camera image coordinates can then be used to accurately map your virtual image unto the correct position within that video frame. This is repeated for every frame thus producing the effect of a virtual object superimposed on our real environment as captured by the camera.

4 Image Reprojection

Essential to our approach is the ability to project our overlay onto the required portion of the real world. It is necessary to obtain the coordinates of a point (x' , y') of one frame given its corresponding coordinates (x , y) in another frame. We need this ability to reproject our augmented overlay image from one position within the current frame to another set of coordinates in some future frame providing seamless augmentation under camera motion. It also enables us to determine the appropriate region within the background to be occluded.

We use ARTag to locate potential marker projections in an image by first finding four sided border contours. This is achieved using an edge detection algorithm which thresholds edge pixels, links them into segments and grouped them into quadrilaterals.

The corner points of these contours are then used to define a homography to create a sampling grid.

An homography is defined in 2 dimensional space as a mapping between points on a ground plane and as seem from one camera to the same point as seen from another camera or another location. When homogenous coordinates are used, it is a linear mapping described by a 3x3 non-singular matrix such that $u' = Hu'$ for some homography H. A homography has eight degrees of freedom, but for every point to point correspondence $u - u'$, two constraints are imposed on H corresponding to the x and y coordinate coordinates of that point. This allows ARTag to specify at least four coordinates as obtained from edge contours to define the appropriate homography. An homography can be represented mathematically as follows:

$$p_a = \begin{bmatrix} x_a \\ y_a \\ 1 \end{bmatrix}, p_b = \begin{bmatrix} x_b \\ y_b \\ 1 \end{bmatrix}, \mathbf{H}_{ab} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$

$$p_b = \mathbf{H}_{ab}p_a \quad (1)$$

where p_a and p_b are the homogenous coordinates representing the original and transformed space respectively.

Once the sampling grid is obtained, it can be used to analyse the interior region to determine the authenticity and identity of the potential marker.

Once the marker is detected we determine the centre of the marker based on the projected coordinates of its four vertices. This is repeated for each frame allowing us to track and update a given pixel position from frame to frame.

5 Background-Foreground Segmentation

In this paper we use a background modeling algorithm based on "An Improved Adaptive Background Mixture Model for Real-time Tracking and Shadow Detection" presented by Trakulpong and Bowden [4] and implemented in OpenCV's library. This model was a further improvement of the one of the more successful background models by Grimson *et al* [13] which used a multi-colour per pixel approach.

This method uses the work presented by Grimson *et al* which show that pixels usually have a bimodal distribution and can thus be actively modeled using Gaussian distribution. Each pixel in the scene is modeled by K Gaussian distributions where the probability that a certain pixel has a value X_n is given by

$$p(x_n) = \sum_{j=1}^K w_j \eta(x_n; \theta_j) \quad (2)$$

where w_k is the weight parameter of the k^{th} Gaussian component. Given X_k as the random variable corresponding to the pixel value for component k , we assume it follows a normal distribution defined as follows:

$$\eta(\mathbf{x}; \theta_k) = \eta(\mathbf{x}; \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_k|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\mu_k)^T \Sigma_k^{-1} (\mathbf{x}-\mu_k)} \quad (3)$$

Different Gaussians are used to represent different colours. The weight parameter w_k models the proportion of time that a given colour assuming A highest probabilistic colours. The model contains an update system to facilitate real time running and also to account for changes in lighting conditions. This works by comparing every new pixel value against existing model components and updating the first matched model component. If no match is found, a new component is added. More details of the theoretical approach can be obtained through reading [4].

The background model is first initialised using the first frame captured by the camera which acts as our background frame. Additional frames are added to the model to provide real time updates of the background as the scene changes. A foreground contour mask is created using the model to highlight foreground pixels based on the current background. This is then used to determine what regions of the overlay need to be occluded.

6 Implementing Virtual Advertisement

6.1 Development Environment

The implementation of the augmented reality sports advertisement replacement application was carried out using a Compaq Presario V2000 laptop, a quite affordable consumer level PC. It ran Microsoft Windows XP Professional Service Pack 2, running on AMD Turion 64 Mobile 1.8GHz processor, 512 MB main memory and 100 Gbyte hard-drive.

To capture the required video image, a Logitech QuickCam Pro 5000 USB camera was used, having a shutter speed of up to 30 frames per second live video, true 640x480 pixel live video and high-quality VGA sensor using RightLight technology.

Capturing resolution was set to 320x240 and using OpenGL the image was stretched to 1024x1024 but fitted in an 800x600 OpenGL generated window.

Microsoft Visual C++ 8 part of the Microsoft Visual Studio 2005 software development suite was used for programming the application.

6.2 The Application

The advertisement replacement application works by loading and augmenting ordinary digitally designed

jpeg, gif or bitmap images into a real environment such that they appear as part of the environment. In this way advertisements can be designed by image experts and directly rendered into a real environment using the program.

As previously mentioned, ARTag and OpenCV libraries were all used to implement the program. All these libraries are based on the C programming language and so by extension this was the main language of development.

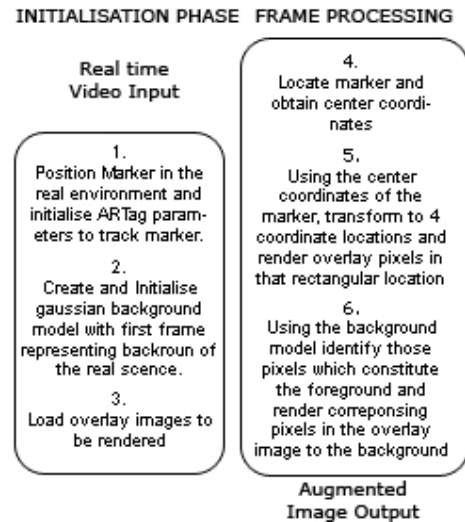


Figure 5: Overview Flowchart of Application System.

6.2.1 Capturing and rendering the

The video stream from the QuickCam Pro 5000 USB camera was opened using OpenCV's cvCaptureFromCAM function and individual frames captured using cvQueryFrame.

Importing Images

OpenCV's cvLoadImage is used to load the advertisement banners, a series of images (jpeg, bmp and gif) located in the project directory. These images were designed and created using Adobe Photoshop CS2 and exported in the required format. The images contained no specialised content related to the performance of the system, but simply represented simulated commercial content.

Tracking Marker

A single ARTag marker was used to demonstrate the application. The marker was printed on ordinary paper and placed against a solid cardboard surface to act as a place holder and to prevent distortion of marker surface. This was then positioned on a wall surface at the point where the advertisement would be rendered. The surface chosen was an arbitrary point in my room so as to represent nature conditions under which such content would usually be rendered. Apart

from marker placement, no special constraints were imposed on the real environment.

7 Results

This represented the actual running of the application and consisted of a series of tests aimed at evaluating the performance of the system. All tests were conducted without any prior special camera calibration techniques and within the development environment as described above.



Figure 6: Accuracy of ARTag markers under occlusion conditions.

The first test consisted of waving my hand at various marker positions to test the accuracy of ARTag marker detection particularly under near marker occluded conditions. As shown in figure 6, these tests were very successful as ARTag proved to be very accurate. The second test involved continuously augmenting the overlay advertisement images in the background to ensure that the system was correctly and accurately performing augmentation. This was analysed for accuracy in terms of marker detection, and accurate calculation of overlay positioning coordinates as shown in Figure 7 below.



Figure 7: Augmentation under motion and changing overlays.

The third series of tests involved performance of the background foreground segmentation algorithm. This involved running the background modeling algorithm on its own and waving a pen object in front of the camera. The foreground image was captured displaying the background in black and all foreground objects in white.



Figure 7: Background-Foreground segmentation using a waving pen.

Finally the entire system was tested running the overlay program within the context of a stationary background and moving hands within and out of the overlay region to observe occlusion of the advertisement. The ability to detect and address occlusion under conditions of camera motion was also tested and it was observed that the background model was able to update itself fairly rapidly under motion. This caused those regions of the overlay image which were rendered occluded under motion to be once again unoccluded once the model had been updated. Updates were however still so slow that full and seamless occlusion was not always obtained.

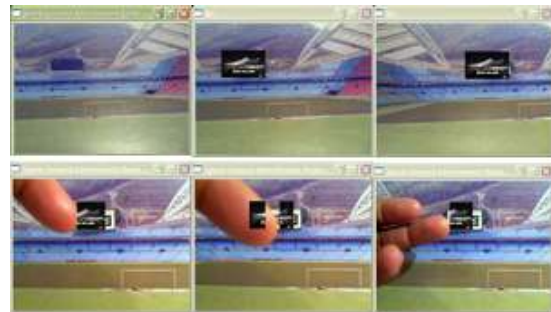


Figure 8: Accuracy of ARTag markers under occlusion conditions.

8 Conclusion and Future Work

We have presented in this paper a method for augmenting 2D virtual content into real time video feed using an off the shelf affordable web camera, ARTag fiducial markers and the OpenCV library. Other than that presence of a physical marker, the method used has imposed no restrictions or constraints on the real world rendering the augmentation in real time, under the camera motion and often in a very seamless way. We have furthermore improved on the realistic rendering of the augmentation by implementing background-foreground subtraction to identify foreground objects and render them unoccluded in the presence of the augmentation. This has been achieved not at the expense of significant modeling of the environment. Recent applications seek to combine ARToolkit with modeling libraries such as OpenSceneGraph (OSG)

[14] while other systems develop quite complex models of the environment. Our approach has demonstrated also that while ARToolkit is widely used, ARTag can also be used successfully to provide accurate tracking for such systems and achieve higher accuracies.

Further work should be considered in using multiple markers to render multiple adverts as well as improving on the virtual realism of the augmentation, possibly through anti-aliasing, lighting effects and alpha maps. The approach used also has some limitations related primarily to the rendering and occlusion of foreground objects. These include marker occlusion specifically under conditions where multiple objects are rendered. Successful occlusion handling also depends on the background-foreground segmentation which takes some time before the background model is accurately updated. This leads to inaccuracies in occlusion detection and poses a problem under rapid camera movement.

The limitations and possible improvements mentioned above, pave the way forward for this approach which can be further tailored to achieve better performance.

9 References

- [1] M. Fiala, "ARTag, an improved marker system based on ARtoolkit", National Research Council Publications NRC47166/ERB-1111, 2004.
- [2] M. Fiala, "ARTag revision 1, a Fiducial Marker System Using Digital Techniques", National Research Council Publications NRC 47419/ERB-1117, 2004
- [3] G. Wel, S.C. Romano and R. Lee, "Using adaptive tracking to classify and monitor activities in a site" in *Proceedings 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No. 98CB36231)*. IEEE Comput. Soc. 1998. 1998.
- [4] P. KaewTraKulPong, R. Bowden, "An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection" in *2nd European Workshop on Advanced Video-based Surveillance Systems*. Kingston upon Thames, 2001.
- [5] Wikipedia, The Free Encyclopaedia, "Home Page" <http://en.wikipedia.org/wiki/>, visited on 15/9/2006.
- [6] C.B. Owen, J. Zhou, K.H. Tang and F. Xiao, "Augmented Imagery for Digital Video Applications", in *Handbook of Video Databases: Design and Applications*, B. Furht and Marques, Chapter 13, CRC Press, Boca Raton, Florida, June 2003.
- [7] R. Azuma, Y. Bailot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre, "Recent Advances in Augmented Reality", *IEEE Comp. Graph. & App*, vol. 21, no. 6, pp.34-47, November 2001.
- [8] T.D. Kammann, "Interactive Augmented Reality in Digital Broadcasting Environments", Diploma Thesis, Universitat Koblenz – Landou, November 2005. https://www.uni-koblenz.de/FB4/Institutes/ICV/AGMueller/DiplomaTheses/index_html
- [9] D. Joele, "Development of an Augmented Reality system using ARToolkit and user invisible markers", Delft University of Technology, Valencia, May 2005. <http://graphics.tudelft.nl>.
- [10] S. Vogt, A. Khamene, F. Sauer, A. Keil, and H. Niemann, "A High Performance AR System for Medical Applications", *Proc. IEEE Int'l Symposium on Mixed and Augmented Reality (ISMAR)*, pp 270-271, October 2003.
- [11] H.L. Wang, K. Sengupta, P. Kumar, and R. Sharma, "Occlusion handling in augmented reality using background-foreground segmentation and projective geometry", *Presence: Teleoperators and Virtual Environments*, volume 13, issue 2, pp 264 – 277, June 2005.
- [12] P. Milgram, H. Takemara, A. Utisumi, and F. Kishino, "Augmented Reality: A Class of Displays on the Reality-Virtuality Continuum", *Proceedings of Telematcher and Telepresence Technologies*, pp 282-292, 1994
- [13] W. Grimson, T. Lazono-Perez, W. Wells, G. Ettinger, and R. White, "An automatic registration method for frameless stereotaxy, imaged guided surgery, and enhanced reality visualisation", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp 330-436, 1994.
- [14] M. Haller, W. Hartmann, T. Luckeneder, and J. Zauner, "Combining ARToolkit with scene graph libraries", *Augmented Reality Toolkit, The First IEEE International Workshop*, Darmstadt, Germany, September 2002.

A Hybrid Approach For Tracking Eye Pupils

M. Schoo, R. Green

University of Canterbury, P.O. Box 4800, Christchurch 8140, New Zealand.

Email: msc61@student.canterbury.ac.nz

Abstract

Applications requiring the accurate identification of eye pupil position in a two dimensional image can be found in many areas, ranging from gaze based computer interfaces to motion capture. Many possible solutions have been put forward to this problem including approaches involving Haar cascades, thresholding, Hough transforms, templates and pattern matching. We present a hybrid pupil tracking algorithm combining Haar face detection, anthropometric localisation, pattern matching and row vs column intensity histograms. We test the performance of our approach on a 285 frame video displaying a variety of gaze directions. Our hybrid approach performs well, resulting in a very low pixel error. Some steps in our technique show similarity with work done by Jin et al but were developed without knowledge of this work.

Keywords: Eye Tracking, Pupil Tracking, Haar Cascades, Histogram, Pattern Matching

1 Introduction

Applications requiring the accurate identification of eye pupil position in a two dimensional image can be found in many areas, ranging from gaze based computer interfaces [1] to motion capture. Many possible solutions have been put forward to this problem including approaches involving Haar cascades, thresholding, Hough transforms, templates and pattern matching.

Kapoor and Picard [2] present pupil tracking by using the red-eye effect. This uses an IR sensitive camera viewing IR light from an LED shining on the face. They publish a high degree of accuracy but such systems suffer from requiring specialised equipment.

Tian et al [3] propose a dual state, convergent tracking approach to determine many eye parameters. While their approach is published to determine accurate eye features in 98% of test frames, a frame rate of only three frames per second is achieved.

Many other techniques exist. The authors direct the interested reader to [4, 5, 6, 3, 7].

We present a hybrid pupil tracking algorithm combining Haar face detection, anthropometric localisation, pattern matching and row vs column intensity histograms. We test the performance of our approach on a 285 frame video displaying a variety of gaze directions. Our hybrid approach performs well, resulting in a very low pixel error.

2 The Hybrid Algorithm

We present an algorithm, created out of several techniques, that recursively narrows the region of interest from the full frame to the pupils. Figure 1 outlines our technique and the separate steps are outlined in more detail in the sections below. We begin by using Haar-like features to detect the face region (section 2.1). Known anatomical proportions of the face allow us to further narrow the region of interest to the area around the eyes (section 2.2). Pattern matching using grayscale eye like images is now used to identify the left and right eye separately (section 2.3). Finally, we use thresholding and vertical and horizontal histograms to find the location of the pupil within the eye (section 2.4).

2.1 Haar Face Detection

The use of statistical cascade classifiers based on Haar-like features have a long history in object detection, particularly in face detection [8, 9, 10, 11, 4]. So widespread is their use that many computer vision APIs, such as OpenCV [12], ship with example Haar classifiers and the functions to access them. The OpenCV implementations of Haar classifiers were used by our system.

Such techniques involve training a cascade of boosted tree classifiers via several thousand positive sub-images (often 24x24 pixel) and negative images. In the OpenCV implementation, simple features are described by a number of templates as shown in figure 2 and described in

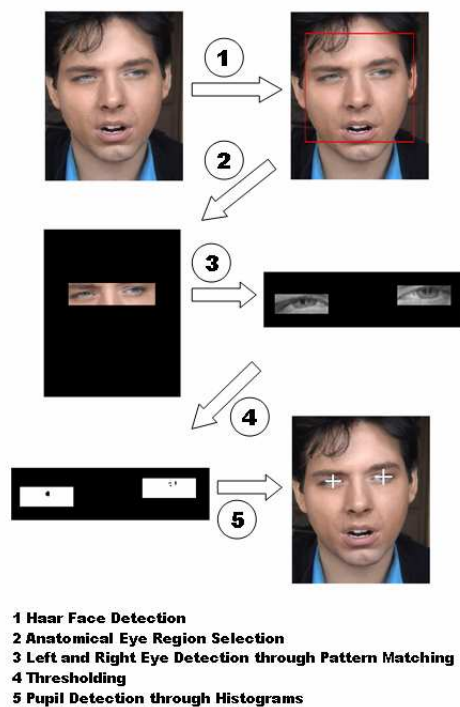


Figure 1: Hierarchical Methodology for Pupil Detection

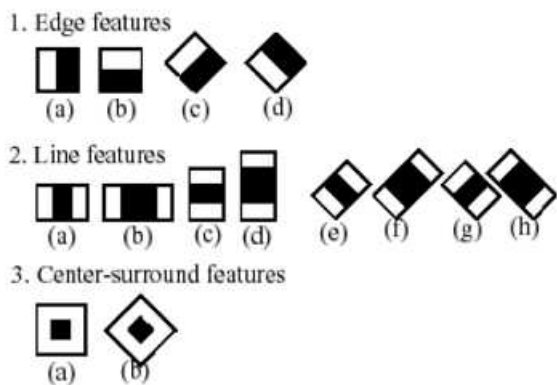


Figure 2: Extended Set of Haar-like Features used in OpenCV's Implementation of Face Detection (taken from [9])

[9]. Detection involves moving the classifier in a sliding window around the image in an attempt to find regions of the image that match the classifier.

Haar cascade face detection makes up the first step of our algorithm. This step narrows the region of interest in the frame to the face of the subject (see figure 1 step 1). Subsequent steps below only consider this localised region.

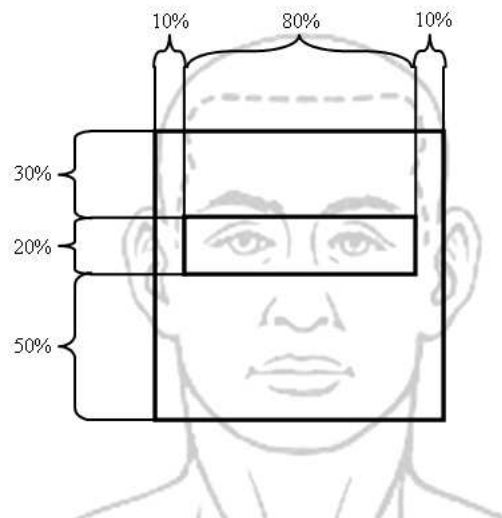


Figure 3: Anatomical percentages used to isolate the eye region.

2.2 Anatomical Eye Region Selection

An assumption in our system, that the above Haar cascade relies upon, is that the user is facing the camera with reasonably little rotation around the three primary axes. Such an assumption allows us to further narrow the region of interest using known anthropometric proportions of the eyes within the face region. Figure 3 shows the proportions used to isolate the eye region. Several different ratios were considered. It was found that the 1:8:1 horizontal and 3:2:5 vertical ratios defined the smallest eye region that robustly segmented the eye region given an image with a successful Haar based face segmentation.

2.3 Individual Eye Detection Through Pattern Matching

By dividing the eye region (see 3rd image of figure 1) in half with a vertical line we have relatively (to the original image) small regions of interest containing the left and right eye. If we make the assumption that these regions contain only the eye we may step directly onto the histogram technique described in section 2.4. However, in many cases these eye regions also contain part or all of the eyebrows. Since, commonly, eyebrows are dark in colour, if we do not remove them from the region of interest the histogram technique will fail.

We further narrow the region of interest for the left and right eye by using a generic, low resolution, grayscale eye image. An attempt is made to find the area in the eye region which best matches this pattern. The process takes place with a grayscale version of the original frame.

Figure 4 shows the images used as patterns throughout investigations of this technique. Pattern 4(b) was eventually used in all experiments.

We wish to find a rectangle of pixels, in the $s \times t$ eye region I, that best matches the $n \times m$ pattern P. If we define the top left corner of this best match to be C and put $X_{x,y}$ as the pixel of some image X intersected by the x^{th} column and y^{th} row, then the best match in the eye region is defined as $C = I_{x,y}$ where $0 \leq x \leq s-n$ and $0 \leq y \leq t-m$ and $E(x,y)$ is the minimum across all x,y as given by equation 1.

$$E(i,j) = \sum_{k=0}^n \sum_{l=0}^m |I_{i+k,j+l} - P_{k,l}| \quad (1)$$

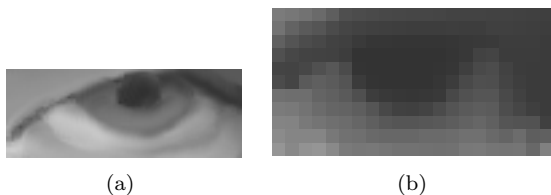


Figure 4: Examples of grayscale eye patterns.

2.4 Pupil Detection Through Row vs Column Histograms

Having narrowed the region of interest to two small rectangles representing the left and right eye (see image 4 of figure 1), we consider a technique for finding the pupil. As the left and right eye regions do not contain eyebrows, we may make the assumption that the pupil is the darkest region in the eye. As such, we threshold the image into a binary image so that the pupil area appears black or 'on', while most other areas are white or 'off'. We count the number of 'on' pixels in each row and column and consider the pupil to be at the intersection of the column and row with the most 'on' pixels. Formally, given an $n \times m$ eye region R, we define the pupil location P as;

$$P = \left(\operatorname{argmax}_i \left(\sum_{j=0}^m R_{i,j} \right), \operatorname{argmax}_j \left(\sum_{i=0}^n R_{i,j} \right) \right)$$

This concept is clarified in figure 5.

3 Results

We tested our implementation of this hybrid approach over 285 frames of a sample 320 x 240 avi video on a 2.8GHz Intel Pentium 4 with 448MB

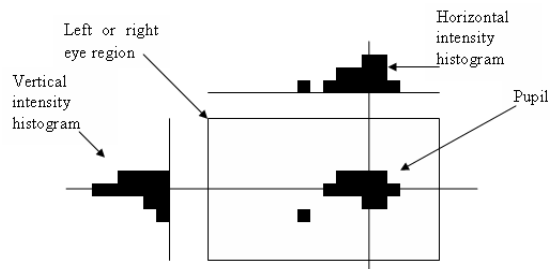


Figure 5: Vertical and horizontal histograms showing the location of the pupil.

Average pixel error	3.19
Maximum pixel error	12.5
Minimum pixel error	1
Average Frames Per Second	4.75

Table 1: Results for experiment on 285 frame test video.

of RAM. The video showed a subject moving their eyes left, right, up and down as well as moving their head left, right, up and down with respect to the camera. No artificial lighting was used in the recording of the video though the room was well lit with natural sunlight diffused by cloud. The pupil locations in all 285 frames were manually recorded and these values were compared with the values returned by our algorithm. Examples of correct and incorrect pupil segmentations are shown in figures 6 and 7 respectively.

Results for the test video are shown in table 1. We see that the algorithm achieves an excellent average error of only 3.199 pixels, compared to the average width of the eye in the frame of 21 pixels. The maximum error measured between actual pupil and detected pupil was 12.5 pixels. Such a low error is due to the hierarchical nature of the algorithm. That is, should the final stage of the algorithm be unsuccessful at locating the pupil, a successful eye region detection by the previous step will have reduced the possible error.

While the number of frames per second is higher than some approaches, it is still too low to be considered for real time applications of either motion capture or gaze based input. The low frame rate is due primarily to the Haar face detection.

4 Conclusion

Applications requiring the accurate identification of eye pupil position in a two dimensional image can be found in many areas, ranging from gaze based computer interfaces to motion capture. We present a hybrid pupil tracking algorithm combining Haar face detection, anthropometric localisation, pattern matching and row vs column intensity



(a)



(b)



(c)

Figure 6: Examples of successful pupil location.

histograms. We tested our implementation of this hybrid approach over 285 frames of a sample 320 x 240 avi video. The system achieved an average pixel error of 3.199 pixels and ran at a speed of 4.75 frames per second.

Further development will tune this system, addressing its limitations. Most notably, these include a frame rate that makes real time tracking difficult and high error results when the user blinks. Improving the speed of the Haar face detection algorithm (the slowest part of the system) and including a technique to detect a blink state will address these issues.



Figure 7: Example of unsuccessful pupil location

5 Acknowledgements

The first author would like to thank Amol Malla and Kon Zakharov for their invaluable help in the formulation of this algorithm and Jennifer Schoo for her comments.

References

- [1] G. Bradski, "Computer vision face tracking for use in a perceptual user interface," *Intel Technology Journal*, no. 2, 1998.
- [2] A. Kapoor and R. W. Picard, "Real-time, fully automatic upper facial feature tracking," in *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pp. 8–13, May 2002.
- [3] Y. Tian, T. Kanade, and J. F. Cohn, "Dual-state parametric eye tracking," in *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition 2000*, (Washington, DC, USA), p. 110, IEEE Computer Society, 2000.
- [4] K. S. Jin, S. G. Cho, J. S. Lee, and J. J. Hwang, "Real-time pupil detection based on three-step hierarchy," in *Signal Processing, 2004. Proceedings. ICSP '04. 2004 7th International Conference on*, vol. 2, 2004.
- [5] M. Covell, "Eigen-points: Control-point location using principle component analyses," in *In Proceedings of the 2nd international Conference on Automatic Face and Gesture Recognition (FG '96)*, (Washington, DC, USA), p. 122, IEEE Computer Society, 1996.
- [6] V. Uzunova, "An eyelids and eye corners detection and tracking method for rapid iris tracking," Master's thesis, Otto-von-Guericke University of Magdeburg, 2005.
- [7] K. T. R. S. Feris, J. Gemmell, "Facial feature detection using a hierarchical wavelet

- face database,” Tech. Rep. MSR-TR-2002-05, Microsoft Research Technical Report, 2005.
- [8] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, pp. 511–518, 2001.
- [9] J. M. R. Lienhart, “An extended set of haar-like features for rapid object detection,” in *Image Processing. 2002. Proceedings. 2002 International Conference on*, vol. 1, pp. 900–903, September 2002.
- [10] A. Kuranov, R. Lienhart, and V. Pisarevsky, “An empirical analysis of boosting algorithms for rapid objects with an extended set of haar-like features,” Tech. Rep. MRL-TR-July02-01, Intel Technical Report, 2002.
- [11] B. S. A. B. S. Romdhani, P. Torr, “Efficient face detection by a cascaded support-vector machine expansion,” in *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 460, 2004.
- [12] OpenCV : Open Source Computer Vision Library, “Home page.” <http://www.intel.com/technology/computing/opencv/index.htm>, visited on 19/9/2006.

Ultrasound Image Segmentation With Multilayer Perceptron-Based Level Sets

M. Mora^{1,2}, C. Tauber¹, H. Batatia¹

¹IRIT-ENSEEIH, Toulouse, France.

²Catholic University of Maule, Talca, Chile.

Email: {mora;tauber;batatia}@enseeiht.fr

Abstract

The class of geometric deformable models, also known as level sets, has brought tremendous impact on medical imagery due to its capability of topology preservation and fast shape recovery. Ultrasound images are often characterized by a high level of speckle causing erroneous detection of contours. This work proposes a new stopping term for level sets, based on the coefficient of variation and a multilayer perceptron, in order to robustly detect the contours in ultrasound images. Successful applications of the MLP-Level Sets to detection of contours on synthetics and real images are presented.

Keywords: Ultrasound images, segmentation, level sets, multilayer perceptron

1 Introduction

The original level sets method has been introduced by Osher and Sethian in [15]. Since its introduction, this technique has been used to solve various problems, such as image enhancement and noise removal [11, 12, 13], and contours detection [10].

Speckle is a multiplicative locally correlated noise. The speckle reducing filters have originated mainly in the synthetic aperture radar community. The most widely used filters in this category, such as the filters of Lee [7], Frost [3], Kuan [6], and Gamma Map [8], are based on the coefficient of variation (CV). For ultrasound images, the use of anisotropic diffusion and CV have been proposed to increase the edge detection effectiveness [19, 20].

The multilayer perceptron (MLP) and the backpropagation algorithm (BP) [17] have been successfully used in classification and functional approximation. An important characteristic of MLP is its capacity to classify patterns grouped in classes not linearly separable. Besides that, it has been shown that a one-hidden-layer perceptron is a universal function estimator [4, 5]. Moreover, there are powerful tools, such as the Levenberg-Marquardt optimization algorithm [2], and bayesian approaches for defining the regularization parameters [9], which enable the efficient training of MLP.

Our work proposes a level set method based on an original stopping term. We create a specific edge stopping term from the weight function of the Tukey's biweight error norm. This term adopts the

coefficient of variation instead of classical gradient for a more robust edge detection. We enhance the performance of the stopping term by estimating the scale parameter automatically and designing a multilayer perceptron trained to differentiate homogeneous areas from edges. This supervised method is designed for segmentation of sequences of ultrasound images. It allows easy robust detection of the contours in subsequent images of the same sequence.

The outline of the paper is as follows. Section 2 describes the level sets principle and its ineffectiveness to segment ultrasound images. Section 3 develops a CV based stopping term robust to speckle. In section 4 we present the MLP based stopping term. Section 5 contains the results on synthetic and real ultrasound images. Finally, section 6 provides some conclusions and perspectives.

2 Shape modeling using level sets

Shape modeling using a level set approach considers a closed curve $\delta(t)$ moving in the plane, where $\delta(0)$ is the initial curve. The curve is represented implicitly via a Lipschitz function. The main idea is to embed this propagating curve as the zero level set of a higher dimensional function $\Phi(\delta, t)$ [10]. The equation representing the motion of the surface $\Phi(\delta, t)$ in the normal direction of the propagating curve is:

$$\frac{\partial \Phi}{\partial t} + F|\nabla \Phi| = 0, \quad (1)$$

where F is the propagation speed function. For certain forms of F , equation (1) reduces to a standard Hamilton-Jacobi equation. The speed function is defined by two terms:

$$F = F_A + F_G, \quad (2)$$

where F_A represents a constant advection term that will force the curve to expand or contract uniformly based on its sign. The second term F_G depends on the geometry of the curve and acts to smooth out high curvature regions. For details on F_A and F_G see [10].

In order to stop the evolution of the curve at the edges, a function of the image gradient is classically used. This stopping term g is defined as follows:

$$g(\nabla I) = \frac{1}{1 + |\nabla(G * I)|^p}, p \geq 1. \quad (3)$$

The term $G * I$ in (3) is the convolution of the intensity image I with a gaussian filter G . The function g has values that are close to zero in regions where the gradient of the image is high, and values that are closer to one in the homogeneous regions.

The traditional edge stopping term g based on gradient has disadvantages. First, the stopping function is never exactly zero and the moving curve may cross the boundaries of the object. In addition, gradient-based edge detection is not adapted to speckle. In images affected by multiplicative noise, the gradient detects outlier contours, resulting in many false positive. This work addresses these issues by using a stopping term based on the CV and an MLP.

3 A stopping term based on the coefficient of variation

The techniques used to reduce the multiplicative noise in radar images use the CV to characterize the noise. The CV, noted ξ can be estimated as:

$$\xi^2 = \frac{\text{var}(I)}{\bar{I}^2}, \quad (4)$$

where $\text{var}(I)$ is the variance of the intensity image and \bar{I} is the mean.

The local version γ of the CV calculated in the vicinity of a pixel $s = (i, j)$ is:

$$\gamma^2(s) = \frac{1}{|\eta_s|} \sum_{p \in \eta_s} \frac{(I_p - \bar{I}_s)^2}{\bar{I}_s^2}, \quad (5)$$

where η_s is a neighbourhood of s , \bar{I}_s is the mean intensity of η_s .

The M-estimator from robust statistics and the edge stopping term controlling the level set evolution can be directly related. In fact the classical stopping term is an adaptation of the weight function of the Fair estimator. To build our new stopping term, we adapted the weight function of the Tukey's biweight error norm [16]. This function neglects the influence of outliers above a predefined threshold. The weight function of the Tukey's error norm expression is:

$$w(x, \sigma) = \begin{cases} \frac{1}{2} [1 - (\frac{x}{\sigma})^2]^2 & \text{if } x \leq \sigma \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Introducing the CV in (6), and considering the edges in images as outliers, allow to totally stop the evolution of the curve on the edges. The expression of the contours detector based on the CV is:

$$g_{CV} = \begin{cases} \left[1 - \frac{\gamma_{i,j}^2}{\gamma_s^2}\right]^2 & \text{if } \gamma_{i,j} \leq \gamma_s \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

where $I_{i,j}$ is the intensity of pixel (i, j) , $\gamma_{i,j}$ is the local CV of that pixel, and γ_s is a scale parameter based on the global CV [19]. Edges correspond to pixels where the values of local CV are greater than the global CV.

We propose to enhance the estimation of the scale parameters γ_s by incorporating a MLP edge detector. The method is described in the next section.

4 MLP-based stopping term

In order to improve the performance of our stopping term, we improve the test in (7) by using an MLP trained to detect contours. The training is made with a method that allows a suitable generalization, to correctly classify patterns that do not belong to the training set. The parameters of the MLP are calculated during the learning process, considering a training set that comes from a single image. We generalize the MLP to correctly detect contours in subsequent images of the same sequence, or in images which have been acquired with similar parameters.

The learning process of MLP supposes the existence of 2 classes in the image: contours and homogeneous regions. The development of our stopping term considers the following stages:

1. Constructing the training set;
2. Training the MLP using the patterns of the training set;

3. Computing the stopping function.

4.1 Constructing the training set

The image of the coefficients of variation, noted I_{cv} , is computed from the original image I . The training set S of the supervised learning neuronal network is:

$$S = \{(p_1, t_1), (p_2, t_2), \dots, (p_i, t_i), \dots, (p_n, t_n)\} \quad (8)$$

where each pattern p_i is a vector that corresponds to the vicinity of a pixel (i,j) of the matrix I_{cv} , and each t_i is the class of the pattern p_i . If p_i belongs to a contour t_i is equal to 0; otherwise it would be equal to 1.

Figure 1 shows the structure of the MLP. The neural network has one output that indicates the class of the inputs. The number of inputs depends on the size of the fixed vicinity.

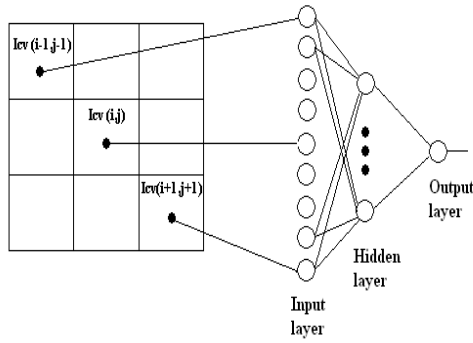


Figure 1: Structure of MLP.

4.2 Training of the MLP

We train the MLP with a Bayesian Regularization Backpropagation (BRBP) [2]. Generalization allows for suitable classification of patterns not participating to the training set. The BRBP provides the number of effective parameters used by the network. This characteristic allows to define the amount of neurons in the hidden layer of the network.

4.3 Computing the stopping function

The MLP described above has been used to design an adaptive stopping term that improves the level set method. The expression of this new stopping term is:

$$g_{mlp} = \begin{cases} 0 & \text{if } \gamma_{i,j} > \gamma_s \\ & \text{or MLP}=1 \\ \left[1 - \frac{\gamma_{i,j}^2}{\gamma_s^2}\right]^2 & \text{otherwise} \end{cases} \quad (9)$$

This term (9) has two interesting properties: it is exactly equal to zero at the edges, and it is robust to speckle.

5 Results

A well known issue with the standard level-sets algorithm is its high complexity. In order to reduce the computation cost, some methods have been proposed as the fast marching approach [18] and the narrow-band approach [1]. We use the latter approach for the experimentations.

5.1 Properties of our stopping term

The effectiveness of our stopping term is illustrated in figure 2. It shows that CV stopping term is no zero on all the contours of the synthetic image, whereas the CV-MLP stopping term in figure 2f is exactly equal to zero on all the extension of contour of the circular object. Figure 2a shows the initial synthetic image without noise; figure 2d shows the initial image affected by speckle; figure 2b shows the stopping term based on the CV; figure 2e shows the zeros of the stopping term based on the CV; figure 2c shows the stopping term based on the CV-MLP; figure 2f shows the zeros of the stopping term based on the CV-MLP.

5.2 Results on synthetic images

We compare the classic stopping term based on gradient, the CV-based stopping term and our original MLP and CV-based stopping term on a synthetic noisy image. The sequence of figures 3a-3h shows the edge detection using narrow-band level set with a gradient based stopping term. The sequence of figures 3i-3p shows the edge detection using a CV based stopping term. The sequence of figures 3q-3x shows the edge detection using our CV-MLP based stopping term. All the figures have as subcaption the iteration number of the evolution process of the curve.

The sequence of figures 3a-3h shows that the moving curve does not suitably detect the objects when the stopping criterion is gradient based. The noise prevents stopping the evolution of the curve at the edges of the objects. The sequence of figures 3i-3p shows that by using a stopping term based on CV, the curve enfold a part of the contour of the circular object. The sequence of figures 3q-3x shows that our stopping term based on an MLP allows to detect all the contours of the object.

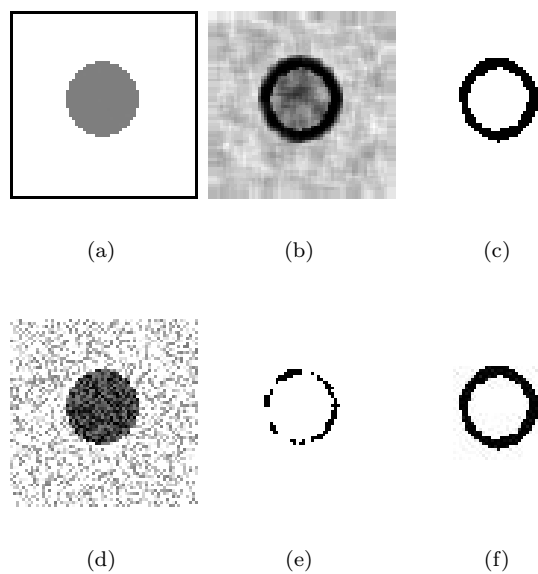


Figure 2: Stopping terms. (a) Original image (b) CV stopping term (c) CV-MLP stopping term (d) Noisy image (e) Zeros of CV stopping term (f) Zeros of CV-MLP stopping term.

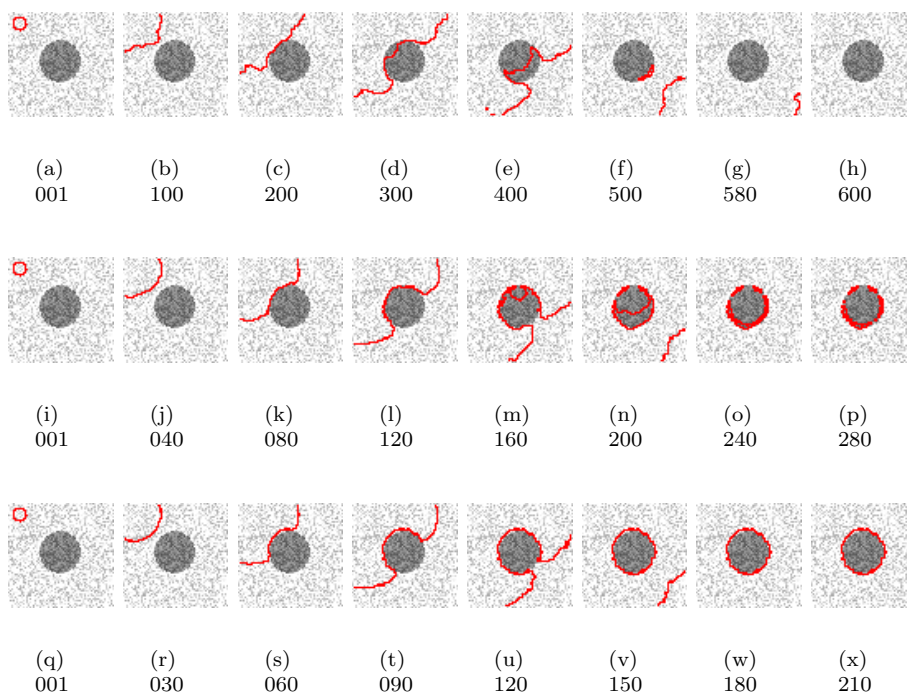


Figure 3: Results on synthetic noisy image. (a-h) Level sets curve evolution with gradient based stopping term. (i-p) Level sets curve evolution with CV based stopping term. (q-x) Level sets curve evolution with MLP-CV based stopping term.

The results presented in figure 3, show that the new stopping term improves the precision of the segmentation. In addition, comparing the number of iterations for the three methods, we observe a better performance of the CV and MLP stopping terms. Less iterations are required to detect the contours in the image.

5.3 Results on real ultrasound images

In this section, we show the performance of our method on an ultrasound intra cavity image. The first row shows results obtained using classical level sets. The second shows the results of our method using the multilayer perceptron classification. These are clearly more precise, our method allows to completely stop the curve on the edge of the heart cavity. The classical method fails to segment the cavity, mainly because of the speckle and the fact that its stopping term never exactly reaches zero.

6 Conclusion

This paper presented a new stopping term adapted to speckle for the level sets algorithm.

The classical edge stopping term based on gradient is not adapted to speckle and never equals zero, making the moving curve pass through object boundaries.

Our stopping term is equal to zero at the edges and it is adapted to speckle. This prevents the moving curve from crossing the boundary of the cavities and increases its precision. In addition, the proposed stopping term reduces the amount of iterations to detect the contours. It brings significant enhancement in the contours detection in ultrasound images using level sets.

The results of this study are promising. Future work will consider the use of an unsupervised model of neural networks (e.g. SOM) for an automated solution of the problem, and the use of other efficient level sets methods for ultrasound image sequences.

References

- [1] Adalsteinsson D. and Sethian J., "A fast level set method for propagating interfaces", *Journal of Computational Physics*, vol. 118, no.2, pp. 269-277, 1995.
- [2] Foresee D. and Hagan M., "Gauss-Newton Approximation to Bayesian Learning", in *Proceedings of the International Joint Conference on Neural Networks*, 1997.
- [3] Frost V., Stiles J., Shanmugan K. and Holtzman J., "A model for radar images and its application to adaptive digital filtering of multiplicative noise", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-4, pp. 157-166, 1982.
- [4] Funahashi K., "On the Approximate Realization of Continuous Mappings by Neural Network", *Neural Networks*, vol. 2, pp. 183-192, 1989.
- [5] Hornik K., Sitchcombe M. and White H., "Multilayer Feedforward Networks are Universal Approximators", *Neural Networks*, vol. 2, pp. 359-366, 1989.
- [6] D. Kuan, Sawchuk A., Strand T. and Chavel P., "Adaptive restoration of images with speckle", *IEEE Transaction on Acoustics, Speech and Signal Processing*, vol. 35, pp. 373-383, 1987.
- [7] Lee J., "Digital image enhancement and noise filtering by using local statistic", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-2, pp. 165-168, 1980.
- [8] Lopes A., Touzi R. and Nezry E., "Adaptive speckle filters scene heterogeneity", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 28, no. 6, pp. 992-1000, 1990.
- [9] Mackay D., "Bayesian Interpolation", *Neural Computation*, vol.4, no.3, 1992.
- [10] Malladi R., Sethian J. and Vemuri B., "Shape modeling with front propagation: a level set approach", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 2, pp. 158-175, 1995.
- [11] Malladi R. and Sethian J., "Image processing: flows under min/max curvature and mean curvature", *Graphical Models and Image Processing*, vol. 58, no. 2, pp. 127-141, 1996.
- [12] Malladi R. and Sethian J., "A unified approach to noise removal, image enhancement, and shape recovery", *IEEE Transactions on Image Processing*, vol. 5, no. 11, pp. 1554-1568, 1996.
- [13] Malladi R. and Sethian J., "Level Set Methods for Curvature Flow, Image Enhancement, and Shape Recovery in Medical Images", *Visualization and Mathematics: Experiments, Simulations, and Environments*, pp. 329-345, 1997.

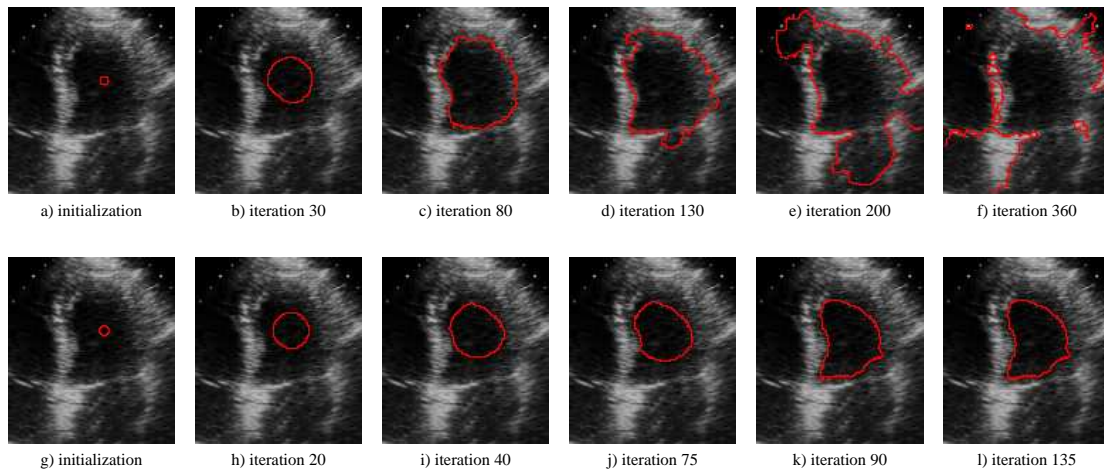


Figure 4: Results on real ultrasound image. (a-f) Level sets curve evolution with gradient based stopping term. (g-l) Level sets curve evolution with MLP-CV based stopping term.

- [14] Mora M., Tauber C. and Batatia H., “Robust Level Sets for Heart Cavities Detection in Ultrasound Images”, in Proceedings of Computers in Cardiology 2005, IEEE Computer Society vol. 32, pp. 235-238, 2005.
- [15] Osher S. and Sethian J., “Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations”, Journal of Computational Physics, vol. 79, no. 1, pp. 12-49, 1988.
- [16] Rousseeuw P. and Leroy A., “Robust regression and outlier detection”, First Edition, John Wiley & Sons, Inc., 1987.
- [17] Rumelhart D., McClelland J. and PDP group, “Explorations in Parallel Distributed Processing”, The MIT Press. vol. 1 and 2, 1986.
- [18] Sethian J., “Level set method”, First Edition, Cambridge University Press, 1996.
- [19] Tauber C., Batatia H. and Ayache A., “A Robust Speckle Reducing Anisotropic Diffusion”, IEEE International Conference on Image Processing (ICIP), pp. 247-250, 2004.
- [20] Yu Y. and Acton S., “Edge detection in ultrasound imagery using the instantaneous coefficient of variation”, IEEE Transaction on Image Processing, vol. 13, no. 12, pp. 1640-1655, 2004.

An Automated System for Microscopy Imaging and Analysis of Histology Slides with an Application in Sheep Meat Morphometry.

V. Hilsenstein¹, P. Jackway¹, P. Allingham²

¹Preventative Health Research Flagship, CSIRO Mathematical and Information Sciences

²Australian Sheep Industry Cooperative Research Centre, CSIRO Livestock Industries
306 Carmody Road, St Lucia, QLD 4067, Australia.

Email: volker.hilsenstein@csiro.au

Abstract

In this paper we present an automated microscopy system that can be used for the scanning and analysis of large collections of histology slides, yielding large mosaic images of entire slides. We present an application of this system in a study of lamb meat texture which aims to link factors such as the growth history and genotype of the sire to morphology of intramuscular connective tissue. For the study, 320 histology slides with stained muscle tissue sections were imaged, and mosaic images of the whole slides were assembled. We analysed the mosaics at different resolutions to detect objects of interest, which were then segmented into muscle tissue, connective tissue and background. We also report initial results for the segmentation of fat cells using seeded watershed transforms.

Keywords: automated microscopy, colour segmentation, meat imaging

1 Introduction

1.1 Automated Microscopy

Many modern laboratory microscopes are equipped or can be retrofitted with motorised components, thus enabling automated microscopy. In addition to enabling researchers in the biological sciences to image and analyse much larger numbers of samples, automation allows for tight control of the imaging parameters and therefore yields images that may be more consistent than those acquired by a human operator. Similar to classical machine vision applications in quality control and parts inspection, consistency in imaging conditions greatly facilitates the subsequent image analysis.

Scanning of entire microscope slides is one of the more common applications of automated microscopy, with a number of commercial products being available for this purpose. Even at relatively low magnifications the size of the resulting image mosaics, often on the order of hundreds of megapixels, poses a data handling challenge. If the purpose of scanning the slides is to further analyse the resulting mosaic images, it is crucial that the image mosaics can be quickly resampled to allow analysis at different scales.

1.2 Aim

This paper reports on the imaging aspects of a project in which we have applied automated microscopy to a study by the Australian Sheep Industry Cooperative Research Centre. The study aims to relate characteristics of the fascicular structure of lamb meat to factors such as the genotype of the sire, nutrition and growth history. It is a follow-on study to the one reported in [1] with a larger sample size.

The automated microscopy system was used to image sheep meat histology sections and automated image analysis was applied to segment muscular and connective tissues. The segmented images will be used for further analysis, which is not described here. Due to the lenses at our disposal, the images were acquired at a magnification that was higher than needed for the texture characterisation. Because this high magnification should in principle allow for the characterisation of fat cells, a capability that might be useful in future studies, we conducted preliminary experiments on segmenting fat cells.

The paper is organised as follows: section 2 presents the sample preparation and gives a brief overview of the microscope hardware. Section 3 details the methods that were used for image acquisition, handling of image mosaics, and

tissue segmentation. The results are presented in section 4. We conclude with a discussion in section 5.

2 Materials and Acquisition Hardware

2.1 Histology Samples

For this study, 320 standard microscope slides were imaged and analysed. On each slide, thin histology sections (4.5 thickness) taken from the muscle *longissimus thoracis et lumborum* from lambs were mounted, with a variable number of sections per slide (between one and four). The size of the tissue sections ranged from $7 \times 7 \text{ mm}^2$ to $16 \times 16 \text{ mm}^2$. An example slide is shown in figure 1.



Figure 1: Example slide with bar code and two sections of muscle tissue.

The histology sections were prepared by the same method as reported in [1], using Wiegert's iron heamatoxylin and Van Geison's stain. This results in a differential stain with different hues for intramuscular connective tissue (dark pink/purple) and muscle fibres (brownish). While there was not much variation in hue between the different samples, there was some variation in the intensity of the stain, both between and within samples. The most probable cause for these variations in intensity are differences in thickness of the sections and differences in absorption of the stain. In some regions, the connective tissue encloses fat cells. These regions can be recognised by their compartmentalised, foam-like structure. The different tissue types are shown in figure 2.

2.2 Hardware

Images for this study were captured using a Qimaging Micropublisher 3.3RTV colour camera mounted on an Olympus BX61 microscope. The microscope is equipped with a motorised stage (Prior H101). In addition, a robotic slide loader (Vision Biosystems SL50) with a capacity of 50 slides was used to automate the slide handling. The setup is shown in figure 3. The Micropublisher camera has a CCD chip with 2048×1536 pixels and uses a Bayer filter pattern for colour imaging.

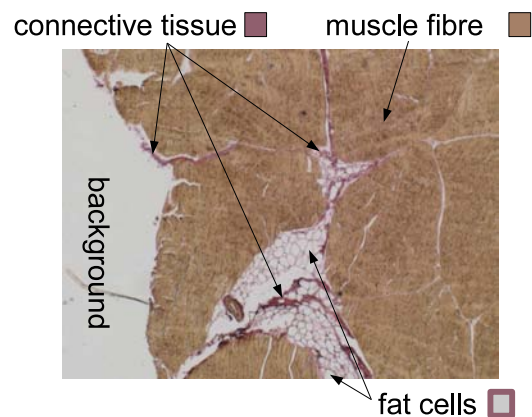


Figure 2: Microscope image of a lamb muscle tissue section showing a single field of view with illustration of the different tissue types.



Figure 3: Microscope (right) with robotic slide loader (left).

3 Methods

3.1 Image Acquisition

The slides were imaged using a 4x lens, the lowest magnification lens at our disposal. This magnification is higher than required for this study, and we scaled down the acquired images by a factor of two to save disk space. However, the high magnification opens up the possibility of automatically detecting and characterising fat cells (see section 3.4.3) which may be of interest for future studies.

Each slide was imaged in its entirety, by moving the stage to 11×24 positions on a regular grid, with a grid spacing of $2.4 \text{ mm} \times 1.8 \text{ mm}$, equal to the field of view of the camera.

The slides were imaged in batches of 50, the maximum capacity of the robotic slide loader. With the acquisition time per slide being just below 30 minutes, we were able to scan each batch in a day

and to complete imaging of the whole set of 320 slides within 7 work days.

To auto-focus we searched for the stage position which maximised the intensity variance across the image. Due to the rich texture of the muscle tissue sections, this simple focus measure worked very reliably.

3.2 Intensity Calibration and White Balancing

To facilitate the colour-based segmentation (see section 3.4) care was taken to keep the colour balance and intensity consistent across all images. To this end we acquired calibration images before scanning each batch of slides.

Using a stack of 10 to 15 images, captured at different positions in empty regions of a sample slide, a calibration image was computed by taking the median value for each pixel and each colour channel across the stack. The median rather than the mean was used to avoid any effect of dust that was present in the otherwise empty fields. The inverse of this calibration image (multiplied by a constant factor) was used to scale each pixel value to the same intensity across the image for each colour channel, thus providing flat-fielding and white balancing.

3.3 Assembling Mosaics using Summed Area Tables

The images were acquired on a regular grid with adjacent, non-overlapping fields of view as described in section 3.1. For each slide, we created a full-resolution mosaic image of the slides by tiling the individual fields of view. We did not employ image registration methods, because alignment errors were small enough to not justify the considerable amount of processing time required. The resulting mosaic images were of the size 18432×11264 pixels. Working with such large images poses some challenges in terms of efficient access and scaling. To address these data handling challenges we developed a storage format which is based on representing the images as summed area tables, reported elsewhere [2]. Figure 4 shows an example mosaic.

3.4 Tissue Segmentation

The tissue segmentation was performed at different scales. First, the position and size of the tissue sections on the slide were determined at low resolution (see section 3.4.1). Second, the sections were extracted at medium resolution and segmented into



Figure 4: Downsampled mosaic image, assembled from 24×11 individual microscopy images of a slide. This mosaic is of the slide shown in figure 1.

muscle fibre and intra-muscular connective tissues (see section 3.4.2). These steps are illustrated in figure 5. Finally, the detection of fat cells was performed at full resolution (see section 3.4.3). Because our detection of objects was based on intensity and the segmentation of tissue types was based on hue, we converted the images from RGB colour space to hue, saturation and lightness (HSL) colour space.

3.4.1 Extraction of Muscle Tissue Sections

To detect the number and position of the tissue sections on the slide we downsampled the mosaic image of each slide by a factor of 10 along each axis. In contrast to the tissue, the white background has very low saturation. Thus we employed thresholding on the saturation channel in HSL colour space to obtain a binary map of the foreground objects. Small gaps in the foreground objects were filled by applying a morphological closing operator with a 7×7 pixel structuring element to the binary map. The resulting foreground objects were further thresholded by area, to exclude small dirt particles and meat debris. Bounding boxes for the remaining objects were calculated and slightly dilated to provide a margin around the tissue sections.

3.4.2 Hue-based Segmentation of Muscle Fibre and Connective Tissue

Based on the bounding boxes of the detected objects, we extracted medium-resolution (4 times downsampled) rectangular sub-windows containing the individual tissue sections from the full-resolution mosaic. To segment the intra-muscular connective and muscle fibre tissue, each extracted image patch was then processed as follows:

1. The foreground regions were determined by setting a saturation threshold and small, isolated objects were removed using an area opening, as described in the previous section for the low resolution image.

- As mentioned in section 2.1, the hues of muscle fibre and connective tissue are visually quite distinct. We used this property for segmenting the two tissue types by applying an empirically determined, fixed threshold to the hue channel of the foreground regions to segment these two tissue types.

3.4.3 Segmentation of Fat Cells using Watersheds

We performed some preliminary work to explore the feasibility of segmenting fat cells. The fat cells appear as regions of background surrounded by connective tissue, as illustrated in figure 2. The compartmentalised structure of the fat cells lends itself to segmentation using a watershed transform [3], which we employed as follows:

- We used hue-based segmentation as described in the previous section to determine candidate regions for the detection of fat cells. The objects classified as connective tissue were filtered by area, to remove small, unconnected regions. A morphological closing operator with a large circular structuring element (40 pixel radius) was then applied to create a region mask.
- We smoothed the lightness channel of the HSL image with a Gaussian kernel (3×3) and detected the local lightness maxima in the candidate regions.
- Using the lightness maxima as seeds, we performed a marker-based watershed transform on the lightness channel in the candidate regions.

4 Results

4.1 Mosaicing

The mosaicing was performed by simple tiling of adjacent fields, without registration. Not using registration led to minor misalignment of objects at the image boundaries. For the purposes of our analysis, this did not pose a problem. Due to the flat-fielding (see section 3.2), intensity differences at the image boundaries were barely noticeable. However, for images covering the meat regions, we noticed a minor colour shift towards the red within a small circular area in the top-right part of each field, despite the calibration. Because of this, a slight colour gridding effect is visible in the mosaic images. The shift to the red was not observed in the background regions, which appeared homogeneously grey after flat-fielding. As the background regions are brighter than the meat regions,

low-resolution overview image

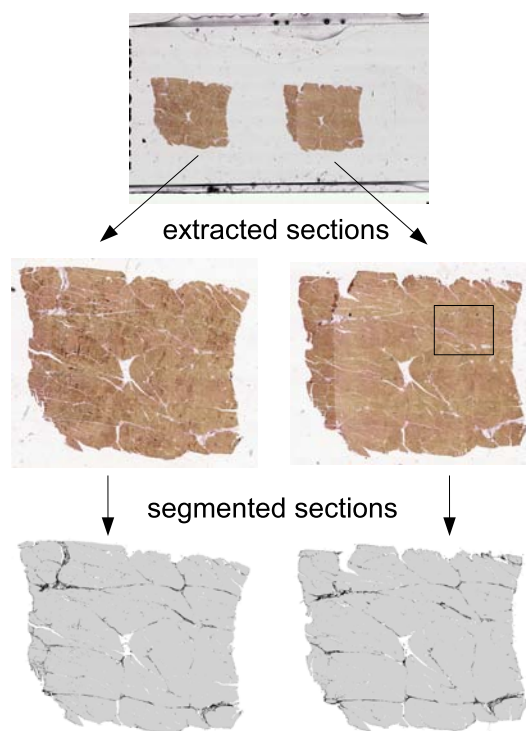


Figure 5: Flow diagram: A low-resolution overview image is used to locate the meat patches on the slide (top). Rectangular image patches are extracted from the original mosaic at higher resolution (middle). The extracted image patches are segmented into intra-muscular connective tissue, shown in black, and muscle fibre, shown in grey (bottom). The region enclosed by the dark square is shown as a close up in figure 6.

this suggests that the balance between the different colour channels of the camera varies with brightness. This may have been due to a defect with the CCD sensor and needs further investigation.

4.2 Extraction of Muscle Tissue Sections

The extraction of tissue sections from the down-scaled mosaic images (see section 3.4.1) worked reliably, with none of the sections being missed. For a small number of slides, excess blobs of the glue used to attach the coverslip were also detected as objects and extracted at high resolution, a problem that could easily be addressed by taking the hue into account at this early stage.

4.3 Tissue Segmentation

The hue-based thresholding (see section 3.4.2) yielded a segmentation of muscle fibre and connective tissue that was generally in excellent agreement with visual assessment by a human

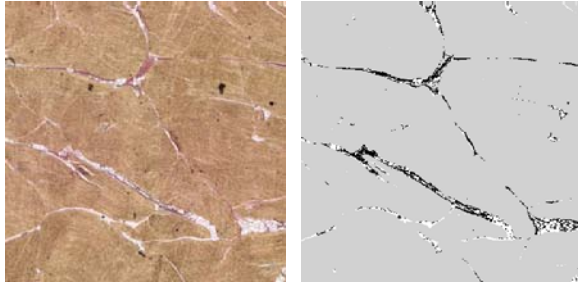


Figure 6: Close-up of the tissue region highlighted in figure 5 (left) and corresponding segmentation results (right).

expert. Figure 6 shows the obtained segmentation for an example section.

Neither the observed intensity variations resulting from differences in stain absorption between different samples, mentioned in section 2.1, nor the slight colour shifts of the camera, mentioned in section 4.1, had a noticeable effect on the segmentation results.

Occasional mis-classification of pixels was observed where dirt was present on the samples and where the glue used for attaching the coverslip had seeped onto the tissue.

4.4 Fat Cell Segmentation

Figure 7 shows the results obtained for the segmentation of fat cells using the method described in section 3.4.3. While many of the cells were segmented correctly, the algorithm also detected some objects that were not fat cells or combined adjacent cells into single objects. These deficiencies can largely be attributed to the method used for detecting markers, which was based on finding local intensity maxima and is not very robust.

5 Discussion

We have described an automated microscopy system for the scanning of histology slides and its application to characterising sheep meat texture. Apart from the obvious advantages of automation, such as reduced manual slide handling and the possibility of analysing a larger number of samples, major benefits result from the high consistency of the images across all slides. As in most machine vision applications, consistent input images greatly facilitate subsequent analysis. In our case, very good segmentation results were achieved using simple thresholding of the images in HSL colour space.

One of the technical challenges of imaging and analysing entire slides is the large size of the resulting mosaic images. For this purpose, we developed a storage format based on summed

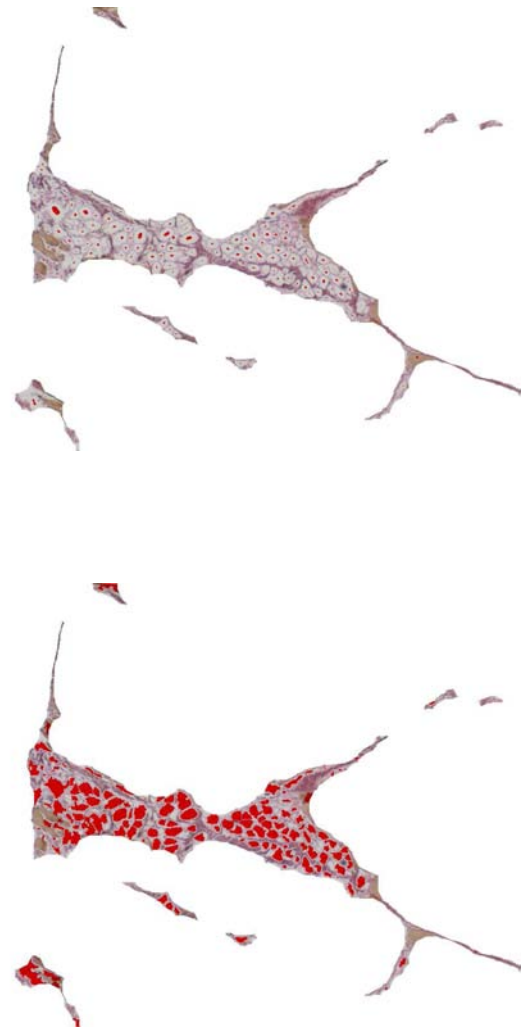
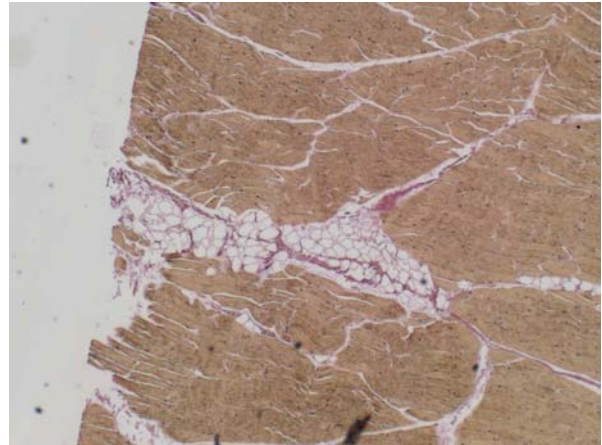


Figure 7: Segmentation of fat cells as described in section 3.4.3: original image (top); candidate regions with seed points overlaid in red (centre); cell regions detected by a seeded watershed transform (bottom).

area tables described in [2], which helped us in analysing the images at different scales.

Even with simple, hue-based thresholding, we achieved good results in terms of separating muscle and connective tissues. By taking neighbourhood information for each pixel into account, the segmentation results could possibly be improved.

The segmented tissue maps (such as the one shown in figure 5, bottom) will be further analysed using stereological methods to derive measures characterising the tissue morphology. The effect of the various factors on the meat texture will be reported in [4].

Characterising fat cells was not a goal of the current study, but our preliminary work on segmenting fat cells indicates that this is feasible. However, the algorithm for segmenting fat cells outlined in section 3.4.3 needs to be improved before being used in practise, for example by employing a more robust method to select seed points, and by splitting segmented objects which have the shape characteristics of two connected cells.

6 Acknowledgements

We acknowledge funding from the Australian Sheep Industry Cooperative Research Centre for this project. We are grateful to CSIRO's Preventative Health Flagship which provided access to the automated microscopy platform.

References

- [1] P. Allingham, G. Gardner, M. Taylor, R. Hegarty, and G. Harper, "Effects of sire genotype and plane of nutrition on fascicular structure of m. longissimus thoracis et lumborum and its effect on eating quality," *Australian Journal of Agricultural Research*, vol. 57, pp. 641–650, 06 2006.
- [2] V. Hilsenstein, "Storing and accessing large images using summed area tables," in *Proceedings Image and Vision Computing New Zealand 2006*, this issue, November 2006.
- [3] F. Meyer and S. Beucher, "Morphological segmentation," *Journal of Visual Communication and Image Representation*, vol. 1, pp. 21–46, September 1990.
- [4] P. G. Allingham, D. Hopkins, V. Hilsenstein et al, "The impact of growth history and sire estimated breeding value on characteristics of intramuscular connective tissue of lamb," *Australian Journal of Agricultural Research*, 2007, in preparation.

Morphological Averaging of Anatomical Shapes Using Three-Dimensional Distance Transforms

Márquez Jorge¹, Patrice Delmas², Isabelle Bloch³ and Francis Schmitt³

¹ Laboratory of Image Analysis and Visualisation, CCADET -UNAM, México

² Dept. of Computer Science, Tamaki Campus, The University of Auckland, Auckland, New Zealand

³ TSI Dept., Ecole Nationale Supérieure des Télécommunications, 46 rue Barrault, 75634 Paris.

Email: jorge.marquez@ccadet.unam.mx

Abstract

Our goal is to obtain robust, morphological averages of anatomical features from a normal population. We present a method for true-morphological, shape-based averaging in three dimensions, consisting of a suitable blend of 3D distance transforms, which code the shape information for N objects, and obtain a progressive average. It is made robust by penalizing, in a morphological sense, the contributions of features less similar to the current average. The morphological error and similarity, as well as penalization, are based on the same morphological paradigm, by defining the squared-difference error in distance transform domain. We present current results for the shape of the human ear in 24 subjects.

Keywords: Morphological average, distance transform, anatomical shape.

1 Introduction

Computer models of anatomical shapes are becoming more representative of a specific population of individuals; they bear complex information providing references for identification of common features which are mapped to and from the atlas and then geometrically and photometrically registered. This allows to make, for example, inter-comparisons, and identification of abnormal conditions. Model-based morphometry is thus a timely strategy for detection of disease-specific variants [1]. Multi-modality atlases allow also to assess pathology and treatment response, when combining several medical imaging techniques [10]. Data to be analysed is multidimensional and complex, hence, atlases provide a reference, a “base truth”, and *a priori* information used for anatomically-driven methods of analysis [2]. Models for the simulation of surgery procedures and augmented reality are also derived from annotated models.

Besides medicine, anatomical models are also used in industry, ergonomics and environmental studies. For instance, the shape alone of the head is considered in the study of power absorption and to design safe communications devices [1]. Concerning dosimetry of hand-held phones, the radio-frequency wave absorption by the human body not only depends on phone terminal positioning, but also depends on anatomical complex features, in particular at the ear and mouth regions. Homogeneous phantoms have been previously used [3], and accuracy improvements

up to 2 mm precision resolution from MRI scans have been made [4]. Representative phantoms from 3D laser-scan acquisitions allow both simulation and experimental analysis of power absorption in a specific population.

Atlas construction must represent a set of individuals and its variability, by considering contributions from each individual according to some similarity with the average features. Shape averages of organs and features are traditionally built by a standard arithmetic averaging of the coordinates of more specific features, such as sets of landmarks and crest-lines [5]. These constitute averages in the *object domain*. A related approach to atlas construction is that of *statistical shape models* or *active shape models* [17], where statistical properties are extracted from landmark information. The statistical and the fuzzy approaches give rise to “soft” models and atlases.

Our goal is to extract a representative instance of a specific facial feature from a large population, or from selected subsets. We developed a morphological, or *shape-based* averaging, generalizing the shape-based interpolation methods from two to N objects. It is an average in the *distance transform domain*, giving rise to “hard” models, where the average itself is an instance of the represented class of features. We also used a robust penalization of outliers (rare variations), and test a simple framework for local penalization, i.e., to penalize local infrequent variations, in order to take into account only the most common sub-features. As a case study, we present results for a shape

average of the external ear, from a set of 24-individuals.

2 Methods

2.1 Laser-scan acquisition protocol

Laser-scan acquisitions were obtained from human heads and processed for the construction of anthropomorphical phantoms. A 3D laser scanner from *Cyberware* [7] was employed; it rotates around the head and obtains distance information. In total, 40 subjects were scanned. Raw data were converted to cylindrical range images, and image processing was applied for 3D-model construction [9] to filter out artifacts and noise, and to locate morphological features in image space and then in object space [8]. A triangulated surface was obtained for rendering and for CAD; it is based on the VRML version 2.0 (*Virtual Reality Modelling Language*) format.

The ear region was then isolated and an average thickness of the ear was estimated on 24 individuals; further details are found in [9]. Head models “with” or “without” ears were built for assessing dosimetry and anthropometry problems. The range-image representation was convenient for other manipulations and measurements, since all 3D information is displayed in the Mercator cylindrical projection as standard grey-level images. A base surface was defined and obtained from boundary information at the auricular-temporal region, using bilinearly-blended Coons patches as described in [9]. Cylindrical projections also simplified the interpolation of the auricular-temporal region, and the 3D geometric registration, as explained next.

2.2 Feature-Based Registration

Morphological tasks such as comparisons and averaging require geometrical correspondence of datasets in a common frame of reference. In a first approach we used a global alignment of principal axes for the bounding box of the auricular-temporal region. A better registration was obtained by extracting robust features of the ears, such as the external crest-lines, available from the depth-range images. In the case of brain structures, the atlas construction process has been adapted to multi-modal imaging [5], and a similar framework is illustrated in Figure 1, where we have integrated our approach considering shape-based interpolation (bottom right) as a deformation. An affine registration is required, since ears vary in size and shape.

We used the *Iterated Closest Point* (ICP) algorithm [11], which is a general purpose, shape-based registration. To better match local features, it was applied on crest line contours from the ear. The steps for registration comprise:

- External ear (pinna) contour (crest lines) extraction.
- Bounding-box extraction in 3D, for each contour.
- Principal axes alignment (preliminary step for ICP alignment).
- Affine registration of contours by ICP algorithm.
- Average geometric transformation.
- Average ROI (bounding-box); average scale calculation.
- Scale normalization of all ears, using the average bounding box.
- Hi-resolution mesh (~2mm), and transformation.
- Voxelization of ROIs. At this step, the registered set is ready for morphological averaging.

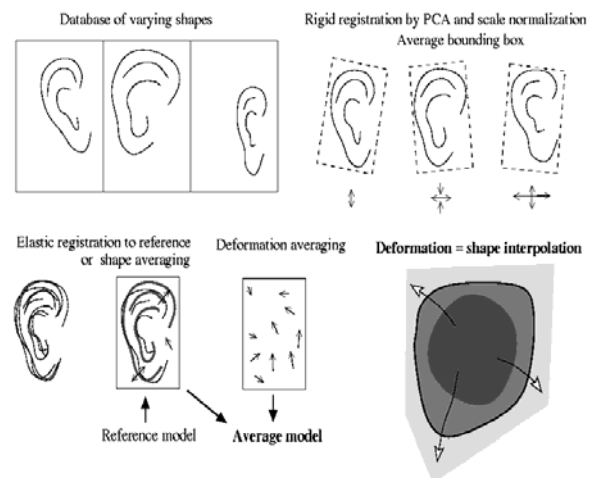


Figure 1: Simplified process for the construction of average models. The concept of morphological average comes from that of morphological interpolation between two similar shapes (bottom right), where the transformation has to deal with the correspondence problem.

Figure 2 shows the feature-based registration, using the crest-lines of the ear borders, and their bounding boxes.

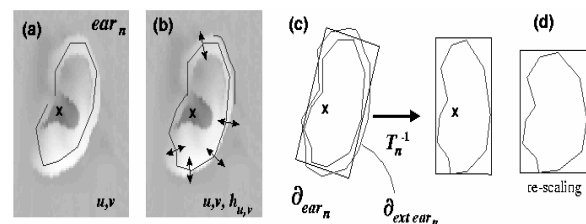


Figure 2. Local, inter-individual registration of the external ear region was based on affine registration of robust features: the crest-lines of the ear borders from the range-images (a,b). Rotation (c) and homogeneous scaling on the 3D bounding box (the figure shows a 2D projection) was done using the average bounding box as reference (d).

Cylindrical coordinates of data from the laser-scanner acquisitions allow to work with 3D surface data as 2D

depth-range images, thus a “2D” bounding box is shown, but the process was done in 3D, before meshing and voxelization. Matching of local features is only approximate, and there may be several deformation paths, or, equivalently, several interpolation paths, as illustrated by the slightly curved arrows of the morphing-shape process of Figure 1, bottom right.



Figure 3. Feature-based registration. (Left) 3D-reference contours (border of the ear) before affine registration. (Right) After homogeneous scaling and PCA registration.

The resulting alignment, for a sample of 5 ear borders, is illustrated in Figure 3. In spite of original registration of the heads during acquisitions, the ear shapes have quite different positions and orientations, besides shape and size variation. An average bounding box provided a frame of reference and a common scale for all ears.

2.3 From morphological interpolation to shape-averaging

Morphological interpolation techniques have been introduced in several fields, from industrial design, to medical imaging, with the common goal of effectively interpolating shape profiles and binary shapes between consecutive image slices, as opposing to signal interpolation in a point-to-point basis on grey-level images. Specific techniques are described in [12], [13], [14]. The concepts of metamorphosis and morphing, are special cases of interpolation of very different, unrelated shapes, with arbitrary correspondence criteria [15], [18]. In inter-slice images, linear interpolation between two similar shapes is usually done as follows.

Let A, B be two discrete objects and let \mathbf{D}_A and \mathbf{D}_B be their discrete, signed, distance fields (also known as distance transforms), then the linear *blending* $\mathbf{D}_\alpha = (1-\alpha)\mathbf{D}_A + \alpha\mathbf{D}_B$, with $0 \leq \alpha \leq 1$ provides a simple way to interpolate an “intermediary” object \mathbf{D}_α between both A, B (these interpolation process is popularly known as “morphing”). The corresponding shape is extracted from iso-surface at a threshold distance zero. The last is also known as a *zero-level set*. The exact average shape corresponds to $\alpha=1/2$. Since it is a linear operation, a third shape C allows conceptualizing a triangle in “shape space” (or more properly, the distance-transform domain) whose

vertices are A, B and C , and the central interpolated shape is a shape-based average. Thus, we define an N -object average. We first list the following definitions:

- \mathcal{V} Digital scene, usually a $N \times M \times L$ volume. The volume \mathcal{V} may be also a *scalar field*, that is, at point (x,y,z) , the quantity $d = \mathcal{V}(x,y,z)$ is a scalar.
- V Discrete object in \mathcal{V} (e.g., digital information of an anatomical structure, represented by an array of scalar or vector values, a mesh structure or by voxels).
- ∂V Boundary of the object V (its discrete surface, either the mesh or its voxelised version).
- $\mathbf{D}(\partial V)$ Signed distance field (a discrete volume with scalar values) associated to boundary ∂V . Note that $\mathbf{D}(\partial V)$ is a scalar field. It is also called the *Distance Transform*. For simplicity we use the notation \mathbf{D}_V .
- $L_{d=d_0}(\mathcal{V})$ Level-set (iso-surface at level $d=d_0$) of a scalar field \mathcal{V} . Note that under certain conditions, at $d=0$ it is possible to define $L_{d=0}$ as an inverse transform: $\partial V = L_{d=0}(\mathbf{D}(\partial V))$.
- $\overline{\partial V}$ A morphological average of N objects V_i , to be defined below.

With the latter notation, we introduce the Euclidean (in the case of Euclidean metric), and the Chamfer (in the case of discrete metric for chamfer-distance transforms) *morphological average* based on the signed distance field associated to the boundaries ∂V_i of each object V_i :

$$\overline{\partial V} = L_{d=0} \left\{ \sum_{i=1}^N \mathbf{D}(\partial V_i) \right\} \quad (1)$$

The zero-level set is the external iso-surface (or boundary) of the average object. $\overline{\partial V}$ is the boundary of a morphological averaged object among several instances V_i , with $i=1, \dots, N$. For precision purposes, we used the Euclidean Distance Transform in all computations.

The averaging can be made “robust” in a statistical sense, when each shape-term of the sum is penalized according to its similarity to the current average, and then recalculating the latter. Let $\{w_1, w_2, \dots, w_N\}$ the normalized set of weights; the starting condition is an homogeneous contribution such that $\sum_N w_i = 1$, and we then define a *robust morphological average*:

$$\overline{\partial V}_R = L_{d=0} \left\{ \sum_{i=1}^N w_i \mathbf{D}(\partial V_i) \right\} \quad (2)$$

An iterative method allows to find an optimal set $\{w_1, w_2, \dots, w_N\}$, from an error measure to be minimized

when comparing ∂V_i with $\overline{\partial V_R}$. The error space is for the moment assumed to be convex (no local minima), but a better analysis needs to be done. A mechanism to compare 3D objects is to use the distance fields of each shape, in order to define a “morphological error”, which can be based on a squared difference, voxel-to-voxel and for each scalar field. For all voxel values $u(x,y,z) \in \mathbf{D}(\partial V_i)$ and its corresponding value $v(x,y,z) \in \left\{ \sum_{i=1}^N w_i \mathbf{D}(\partial V_i) \right\}$,

the volume set $Err^2(V_i)$ is then formed with the following individual, voxel-error values:

$$err^2(x,y,z) = [v(x,y,z) - u(x,y,z)]^2 \quad (3)$$

Note that $Err^2(V_i)$ is a scalar digital scene. Integrating over all (x,y,z) we obtain a global measure of error, permitting evaluation of a natural weight for ∂V_i when averaging at iteration $k+1$. Note that local integration, on any specific region, allows evaluating the contribution to the global error. Thus, weights w_1, w_2, \dots, w_N may be variable, either point to point, locally, or even defined individually for very specific ROIs (a facial feature, for example).

For testing purposes, an incrementing algorithm, modifying each $w_i (i=1, \dots, N)$, at a time, was devised to avoid re-calculating the distance-field average again, by subtracting the contribution of ∂V_i at iteration $k-1$ and updating it with the new value for w_i . A similar approach works for new shapes to be included in the average, which is the essence of a population atlas which becomes more representative with new contributions. In this updating process, the order in which components are added influences slightly the final average result, since old weights are preserved while new weights are assigned according with the actual morphological error, and they may change if a different order is used. To avoid such order dependence, a full population calculation has to be done from the start. We found however useful the incremental approach for fast estimations of the average shape, and to assess error decreasing rates.

3 Results and Discussion

Figure 4 (left) shows sample slices of the 3D distance field for the shape of one single ear, Figure 4 (right) shows sample slices of the averaged distance fields of 24 shapes. For displaying purposes, the unsigned distance field is shown, but averaging is done over signed distance fields. Figure 5 (right), shows a 3D rendering of the resulting average shape, after extracting a triangular mesh from the voxel-representation results, thresholding the average distance field at level $d=0$. There remains the question of validating the average as a feasible human ear, and the method could be refined by constraining the error

minimization while respecting some “ear-ness” criterion. An answer to the question “to what extent the shape-based average is a valid human ear” lies on the size of local mismatches, and is related to spatial sampling limits after according to Nyquist criteria, considering the smallest features to be averaged: the calculated average is an interpolation among real ear features, provided that all their local extrema, or at least the most salient, correspond one-to-one, up to some minimum tolerance. A correspondence mismatch between local extrema (crest lines, in general) implies an invalid local interpolation of at most the size of that mismatch. Non-rigid registration may help to improve correspondence matching. To reproduce the situation of average users of mobile-phones, the acquisitions included a version with the ear collapsed against the skull. The thickness of the corresponding averaged ear, measured by methods reported in [9] was about 6.2 ± 2.5 mm, and agrees well with the measure of this thickness in the Standard Anthropomorphic Model (SAM) [16], obtained by different methods. The present approach also permitted averaging the implicit surface that interpolates the skull, described in [9]. Thus, a morphological-average of the head “without ears” can be obtained, too, for dosimetry studies.

Our method is different from the *active shape models* approach, since we obtain a representative instance of the shape population, a shape-based average which minimizes differences with respect to each instance, and penalizes outlier shapes. Another feature of morphological averaging is that it is easier to incorporate feature-based weightings or other penalization by modifying or steering an associated distance-field potential.

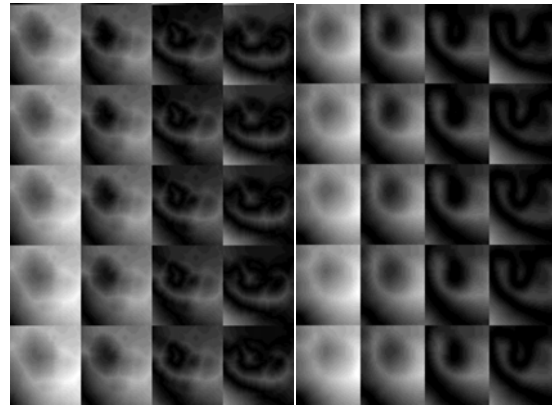


Figure 4 (left) Sample slices of the 3D Euclidean distance field for the shape of one single ear; (right) sample slices of the 3D average distance field from 24 shapes.

At present, we have tested ellipsoid fitting of the head, in order to better align all heads, but the ear positioning do not correspond among individuals and need local registration methods. Another approach to validation of the method is being tested, by directly measuring some features in some of the ears, and then verifying the average measurement in our resulting

morphological average. This has been done manually and we found the average ear thickness (5.9 ± 2 mm) to agree with measures from the widely accepted SAM model [16].

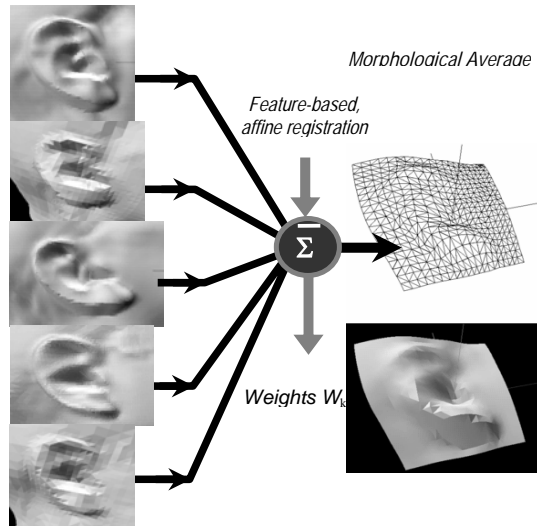


Figure 5. (Left) Elements for the construction of an anatomical, robust, morphological-average of the ear from 24 individuals. Features are the border lines shown in Figure 3, at right. A 3D rendering of the resulting average shape (triangular-mesh representation).

4 Conclusions

An innovative method of shape-based averaging was presented. It blends the distance-field information from several shapes, using distance fields, and a robust implementation was obtained, applying similar ideas to error measures and morphological similarity, calculated also in the distance-field domain, rather than from the shape domain. To test our methods, an average of the ear from a database of 24 human head profiles was obtained. The database was built in voxel format for processing, and in the web-oriented format VRML, for browsing. A base surface was defined and obtained from boundary information at the auricular-temporal region, using bilinearly-blended Coons patches, and the averages of both shapes allowed to measure thickness at various regions of the ear. The ICP algorithm was employed for geometric registration of the bounding boxes, and then for the external border of each ear. A robust, progressive average was then obtained, penalizing at each iteration those ears that gave the largest morphological error from the current average. Averaging of complex and varying shapes has been difficult without a reliable geometric registration, and the implicit Coons surface has alleviated this problem, providing also a baseline for error quantification during registration. It has been observed that ICP registration does not result in good correspondence, and in the present work, a feature-based approach was

incorporated, using the crest-line contours of the external ear, obtaining a more robust registration, before averaging in the distance-field domain. Registration may be based also on chamfer distance fields, a procedure known as “chamfer matching”, and a combination with line feature-based registration is possible by modulating distance fields. Future work includes these enhancements, as well as averaging methods including not only weighted distance fields, but also the fields of non-rigid deformations. Besides anthropometric applications related with atlas construction, a reference model of any feature, obtained by weighted averaging, may be used as a representative instance of a specific human population.

5 Acknowledgements

This work originated from collaborations between the TSI Department of the *Ecole Nationale Supérieure des Télécommunications* (ENST) in Paris, and the *Alcatel Corporate Research Centre*, (*Marcoussis laboratories*). We gratefully acknowledge Drs. Christophe Grangeat and Thierry Bousquet, from the Marcoussis laboratories.

6 References

- [1] P. Thompson, M. Mega, K. Narr, E. Sowell, R. Blanton and A. Toga. *Brain Image Analysis and Atlas Construction*. In M. Sonka and J. M. Fitzpatrick (editors) *Handbook of Medical Imaging*, Vol. 2, *Medical Image Processing and Analysis*. SPIE Press 2000, pp. 1061-1129.
- [2] P. Thompson, and A. Toga. “Brain Warping” in (A. Toga, ed.), *Anatomically-Driven Strategies for High-Dimensional Brain Image*, Acad. Press, 1998, pp. 311-336.
- [3] C. Grangeat et al., “Le rayonnement radiofréquence des téléphones mobiles”, *Alcatel Telecommunications Review*, 4th Quarter (1998), pp.1-8.
- [4] J. Wiart et al., “Calculation of the power deposited in tissues close to a handset antenna using non uniform FDTF”, *Proceedings of the Second World Congress for Electricity and Magnetism in Biology and Medicine*, Bologna, June (1997), Plenum Press.
- [5] G. Subsol, J.F. Thirion, N. Ayache. (1992), “First steps towards automatic building of anatomical atlases.” *Research Report RR-2216*, INRIA Rocquencourt, France, mars 1994.
- [6] J. Márquez, T. Bousquet, I. Bloch, F. Schmitt and C. Grangeat. “Laser-Scan Acquisition of Head Models for Dosimetry of Hand-Held Mobile Phones”, *Bio-*

- ElectroMagnetics Society, 22th Annual Meeting, Technical University, Munich, Germany. Abstract Book. June 9-16, (2000). pp. 123.
- [7] - Inc. Cyberware Laboratory, "4020/rgb 3d scanner with color digitizer", Cyberware, Inc., available at: <http://www.cyberware.com/pressReleases/index.html>, (1990).
- [8] J. Márquez, T. Bousquet, I. Bloch, F. Schmitt and C. Grangeat. (2000b), "Construction of Human Head Models for Anthropometry and Dosimetry Studies of Hand-Held Phones." *Revista Mexicana de Ingeniería Biomédica*. Vol. XXI, No. 4, diciembre de 2000. pp. 120-128.
- [9] J. Márquez, I. Bloch and F. Schmitt. (2003), "Base Surface Definition for the Morphometry of the Ear in Phantoms of Human Heads." *Proceedings of the 2nd IEEE International Symposium on Biomedical Imaging*, Cancún, Mexico, September 15-20, Vol.1, 2003, pp. 541 - 544.
- [10] P. Thompson, R. Woods, M. Mega, A. Toga. "Mathematical/Computational Challenges in Creating Population-Based Brain Atlases". *Human Brain Mapping*, V 9, pp. 811-90, Feb. 2000.
- [11] Besl Paul J and Neil D. McKay. "A Method for Registration of 3-D Shapes". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14 (2), Feb. 1992, pp. 239-256.
- [12] G. J. Grevera and J.K. Udupa, "Shape-Based Interpolation of Multidimensional Grey-Level Images," *IEEE Trans. Med. Imag.*, Vol. 15, No. 6, pp. 882-892 (1996).
- [13] G. T. Herman, J. Zheng and C. A. Bucholtz, "Shape-Based Interpolation," *IEEE Comp. Graph. Appl.*, Vol. 12, No.3, pp. 69-79 (1992).
- [14] S. P. Raya and J. K. Udupa. "Shape-Based Interpolation of Multidimensional Objects", *IEEE Trans. Med. Imag.*, Vol. 9, No. 1, pp. 32-42 (1990).
- [15] D. Cohen-Or, D. Levin and A. Solomovici. "Three-Dimensional Distance Field Metamorphosis". *ACM Transactions on Graphics*, 17, 1998, pp. 116-141.
- [16] – The Standard Anthropomorphic Model (SAM), <http://www.sam-phantom.com/SAMinfo.htm>.
- [17] A. F. Frangi, D. Rueckert, J. Schnabel and W. Niessen, "Automatic Construction of Multiple-Object Three-Dimensional Statistical Shape Models: Application to Cardiac Modeling", *IEEE Transactions on Medical Imaging*, vol. 21, no. 9, September 2002.
- [18] Payne, B.A.; Toga, A.W.; (1992) "Distance field manipulation of surface models." *Computer Graphics and Applications*, IEEE , Vol. 12 Issue: 1 , Jan. 1992. pp. 65 -71.

Image Analysis and Modelling of Disorder in the Myosin Lattice of Vertebrate Muscle

C.H. Yoon, N.D. Blakeley, A. Goyal and R.P. Millane

Computational Imaging Group
Department of Electrical and Computer Engineering,
University of Canterbury
Private Bag 4800, Christchurch, New Zealand
Email: rick@elec.canterbury.ac.nz

Abstract

An Ising model is used to describe the orientational disorder of the filament array observed in electron micrographs of vertebrate muscle cross-sections. The model is evaluated by comparing the second-order statistics from Monte Carlo simulations and data from image analysis of the electron micrographs. For a specific range of temperatures, the results show that the Ising model adequately describes the disorder.

Keywords: Electron micrograph, triangular lattice, disorder, myosin, muscle, Ising model

1 Introduction

Vertebrate muscle fibers contain the contractile proteins myosin and actin which are organised into long thin strands known as myofibrils. The myofibrils exhibit a striated pattern and the repeating unit is known as the sarcomere, which is the basic contractile unit of muscle [1, 2]. The structure of this complex system is studied by both electron microscopy and x-ray diffraction. Myosin filaments pack in a triangular lattice (Figure 1) and can be imaged directly by electron microscopy of thin transverse sections through the so-called bare region. The myosin filaments are roughly triangular in shape and the micrographs are ideal for image analysis. Luther and Squire [3] determined, by visual analysis of micrographs, that in many muscles, such as those from fish, the myosin filaments adopt one of two different orientations that are distributed in a semi-systematic manner. Using these results, they described some general characteristics of the distribution of orientations which were confirmed by a more quantitative analysis by Millane and Goyal [4]. The myosin filament disorder has implications for the contractile mechanism of the muscle, and a good statistical model of the disorder is needed for rigorous interpretation of x-ray diffraction data from muscle fibres. We have previously described a method for automated analysis of electron micrographs to estimate the filament orientations [5, 6]. We describe here analysis of these orientations to obtain a model of the statistical distribution of the orientations.

2 Analysis of Filament Orientations

The presence of a semi-systematic distribution of two different myosin filament orientations was first observed by Luther and Squire in the sartorius muscle of a frog [3]. They noticed that the filament orientations tend to satisfy what they called the *two no-three-alike rules*. Rule 1 is that no three mutually adjacent filaments all have the same orientation. Rule 2 is that no three successive filaments in a row all have the same orientation. Furthermore, they observed that the orientations tend to form what they called a *superlattice*. This refers to the observation that second nearest neighbour filaments (spaced by $\sqrt{2}$ times the fundamental lattice spacing) tend to have the same orientation. The second nearest neighbours form a triangular sublattice that contains one third of the sites of the full lattice (Figure 1). We refer to fundamental (rhombohedral) unit cells of the sublattice whose four vertices contain filaments of the same orientation as *superlattice cells*. Contiguous superlattice cells form regions of superlattice. The particular disorder present leads to peaks in X-ray diffraction patterns from muscle specimens with spacings reciprocal to those of the superlattice [3]. Such peaks would be forbidden in an ordered system. We define the *superlattice content*, denoted f_s , as the number of superlattice cells present divided by the maximum possible number (i.e. if all filaments on the sublattice had the same orientation).

A micrograph from a frog sartorius muscle cross-section is shown in Figure 2a. It shows five myofibrils, each with an ordered packing of myosin filaments. Close-ups of two of these myofibrils are

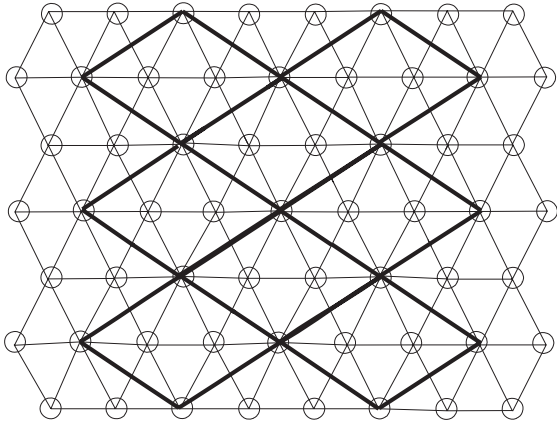


Figure 1: The myosin filament lattice (thin lines) and a sublattice (thick lines).

shown in Figure 2b and c. Myosin filaments (dark regions that are seen to lie on triangular lattices) have approximate triangular profiles. Analyses of the micrographs using methods described previously [5, 6] allow reliable determination of the positions and the orientations of the myosin filaments. The positions are located by convolving a point spread function that incorporates the salient features of the myofibril (i.e. lattice spacing, lattice rotation and filament size) and selecting the regional maxima. The orientations are estimated by fitting a triangular template. A close-up of a part of the micrograph in Figure 2b is shown in Figure 3 with the fitted templates overlaid. A histogram of the orientations for the whole image is shown in Figure 4. Inspection of the histogram indicates that the orientations belong to two populations that are centered approximately 60° apart. As a result of errors in determining the orientations (and probably also of imperfections in the muscle itself) there is a spread of orientations within each population. The underlying distribution of orientations is modelled as a Gaussian mixture consisting of two normal distributions that are wrapped on the interval $(0^\circ, 120^\circ)$. If the histogram for a particular micrograph supports the mixture model (as they do in this case), then each filament is classified into one of the two populations, which are referred to as “up” and “down.” This classification is also shown in Figure 3. Analyses of the distribution of orientations in a variety of micrographs using these methods give results that are consistent with the observations described in the previous paragraph.

In summary then, analysis of a variety of electron micrographs of vertebrate muscle shows that (1) the myosin filaments lie on a triangular lattice, (2)

each filament adopts one of two different orientations (substitution disorder), and (3) the distribution of the two orientations is semi-systematic with neighbouring filaments tending to have opposite orientations and second-nearest neighbours (on a sublattice) tending to have like orientations (superlattice).

3 The Antiferromagnetic Ising Model

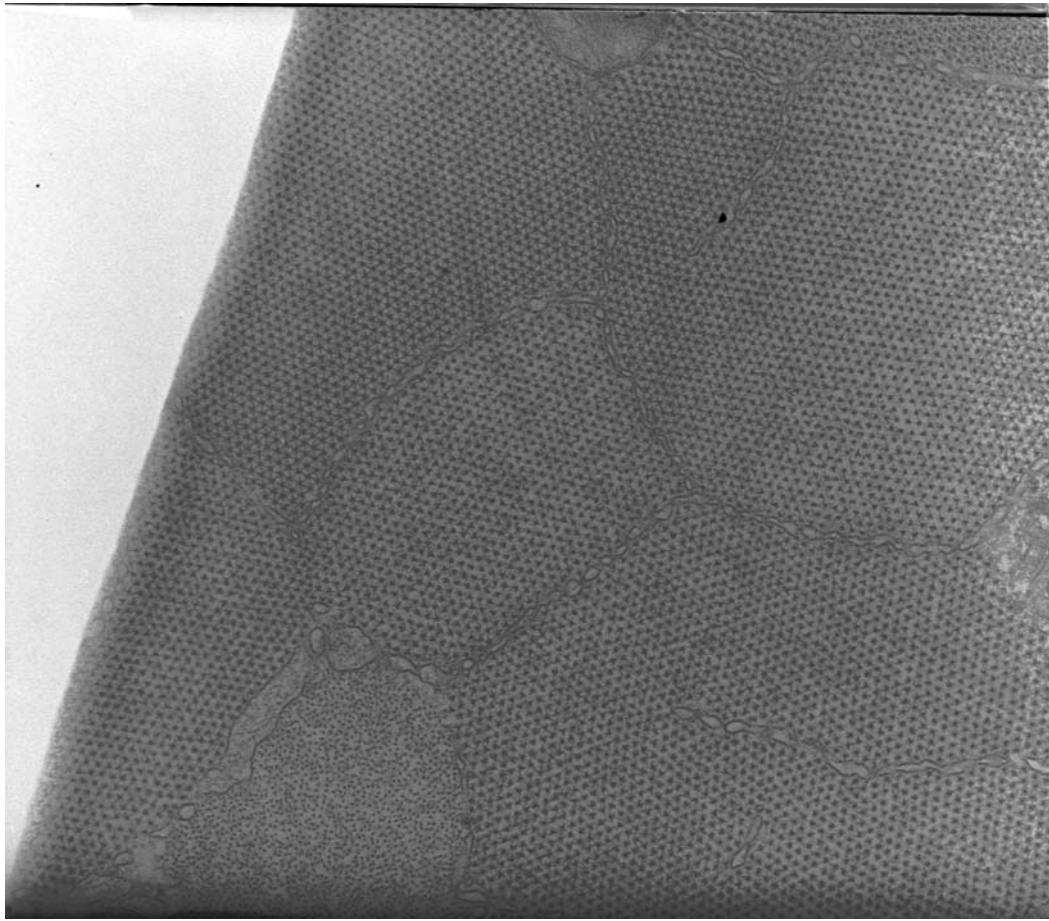
The most important observation from the muscle micrographs is that nearest neighbour filaments prefer opposite orientations. This is analogous to an *antiferromagnetic* interaction in magnet systems where the spins of neighbouring electrons prefer opposite directions as a result of the interaction energy between neighbouring like states (spins) exceeding that of unlike states [7]. We therefore considered a classical two-dimensional Ising model for describing the orientational disorder. In a general statistical mechanical setting, this model is used to describe large lattice systems whose behaviour is driven by the local intersite interaction energies and is a function of the temperature of the system. If only nearest neighbour interactions are present then the antiferromagnetic Ising model on a triangular lattice is *frustrated* since it is not possible to minimise the energy of all pairwise interactions simultaneously on an elementary triangular domain [7, 8]. The result is a large set of minimum energy configurations. These configurations are not random but possess a degree of local order.

The only variable in this model is the (effective) temperature, since the difference in the interaction energy between like and unlike neighbouring spins is just a scaling factor. At low temperatures the system exhibits antiferromagnet ordering and becomes more random at larger temperatures as entropic effects start to dominate.

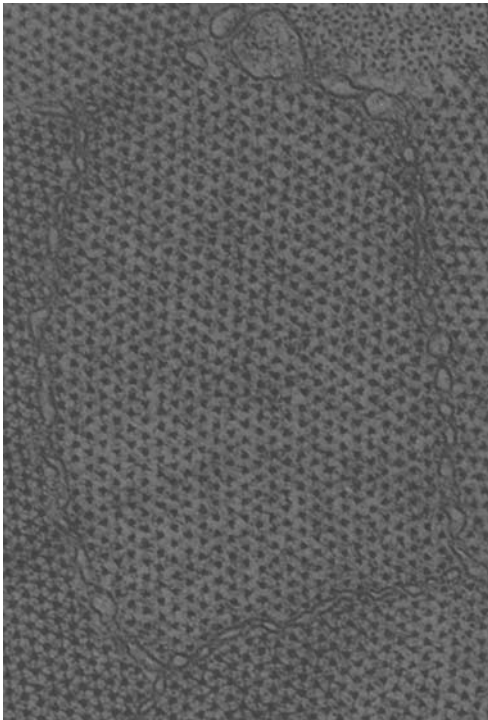
In order to determine if the antiferromagnetic Ising model is a good model of the filament orientations in the myosin lattice, we conducted energy minimisations of the model at various temperatures and compared the statistical properties with those derived from the muscle micrographs. The two systems were compared by comparing their second-order statistics (correlation coefficients) and the superlattice content. The correlation coefficient of the orientations $\rho(\mathbf{d})$ is defined by

$$\rho(\mathbf{d}) = \langle s_{\mathbf{a}} s_{\mathbf{a}+\mathbf{d}} \rangle_{\mathbf{a}}, \quad (1)$$

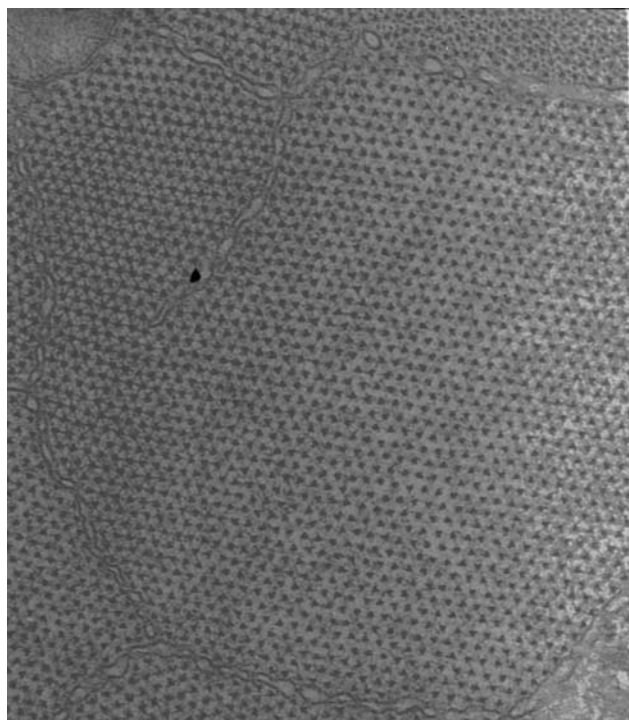
where $s_{\mathbf{a}} = \pm 1$ for up or down orientations at site with vector position \mathbf{a} , $\langle \cdot \rangle_{\mathbf{a}}$ is the ensemble average over \mathbf{a} , and noting that $\langle s_{\mathbf{a}} \rangle = 0$ and $\langle s_{\mathbf{a}}^2 \rangle = 1$. The energy minimisation was conducted using a simple



(a)



(b)



(c)

Figure 2: (a) Electron micrograph of frog sartorius muscle [3], and (b) and (c) close-ups of two myofibrils.



Figure 3: A subimage of the frog sartorius muscle micrograph shown in Figure 2b with the classification of filament orientations denoted by black and white triangles.

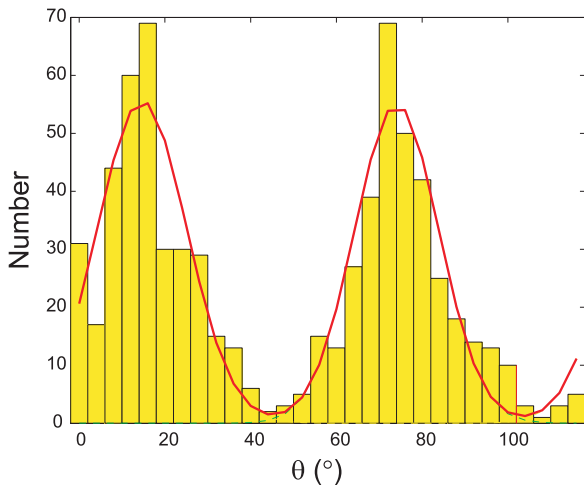


Figure 4: The histogram and mixture model for the image shown in Figure 2b.

Metropolis Monte Carlo simulation as described in the next section.

4 Monte Carlo Simulation

Monte Carlo simulation is a class of computational algorithms for simulating large systems and for locating global minima. We use *single-spin-flip-dynamics* and a simple Metropolis Monte Carlo algorithm [9] to optimise the acceptance ratio. The acceptance ratio, $A(\mu \rightarrow \nu)$, governing a transition from state μ to state ν is

$$A(\mu \rightarrow \nu) = \begin{cases} e^{-\beta\Delta E} & \text{if } \Delta E > 0 \\ 1 & \text{otherwise,} \end{cases} \quad (2)$$

where ΔE is the energy difference between states μ and ν , $\beta = (kT)^{-1}$ where k is Boltzmann's constant and T is absolute temperature. We set the difference in interaction energy between like and unlike neighbouring orientations to unity and $k = 1$ so that T is a normalised temperature.

We used a finite triangular lattice with hexagonal lattice shape with 5000 sites and free boundary conditions for the simulations. For each temperature, a state is chosen at random, its orientation is flipped, and the new state is either accepted or rejected according to the Metropolis criterion. One sweep refers to N attempted flips. The spins are randomly oriented up or down for the initial state. It is important to wait until the system has reached equilibrium at a particular temperature before taking measurements. The equilibration time was estimated by observing how long it took for system parameters to reach a steady state. The correlation function and superlattice content were averaged over samples spaced by twice the decorrelation time. Measurements were typically averaged over 100 states.

5 Results

The Ising model simulation and the distribution of orientations derived from the micrographs are compared by calculating the mean squared difference $e(T)$ as a function of temperature where

$$e(T) = \frac{1}{M} \sum_{\mathbf{d}} [\rho^{data}(\mathbf{d}) - \rho^{sim}(\mathbf{d})]^2, \quad (3)$$

$\rho^{data}(\mathbf{d})$ and $\rho^{sim}(\mathbf{d})$ are the correlation coefficients for the muscle data and simulation, respectively, and M is the number of separations \mathbf{d} . The mean square difference as a function of temperature for Figure 2b is shown in Figure 5. A clear minimum around $T = 0.55$ is evident with a small difference between the measured and calculated correlation coefficients. Correlation coefficients for the muscle data and simulation at $T = 0.55$ versus the magnitude of the separation $d = |\mathbf{d}|$ are shown in Figure 6. Good agreement is evident. The proportion of domains for which Rules 1 and 2 are satisfied are denoted f_1 and f_2 respectively, and are listed in Table 1 as calculated from the micrograph and from the Monte Carlo simulations at $T = 0.55$. The superlattice content f_s is also listed in Table 1. Good agreement is evident in all these quantities.

The myofibril shown in Figure 2c was also analysed and the best match of the correlation coefficients was also at $T = 0.55$. The parameters derived from the micrograph are also listed in Table 1 and are also consistent with the simulation results. Similar results were obtained for micrographs from other specimens with optimum values for temperature in the range $0.45 < T < 0.65$.

Table 1: Various parameters for the micrographs in Figure 2 and the simulation at $T = 0.55$.

Parameter	Figure 2b observation	Figure 2c observation	Monte Carlo simulation
f_1	0.97	0.98	0.98
f_2	0.93	0.93	0.93
f_s	0.40	0.42	0.38

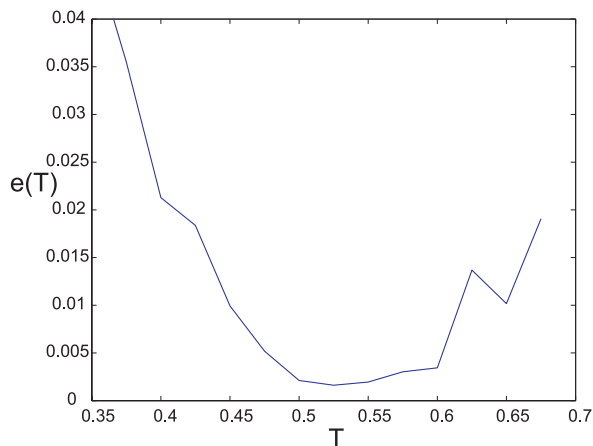


Figure 5: Mean squared difference of correlation coefficients for the micrograph in Figure 2b.

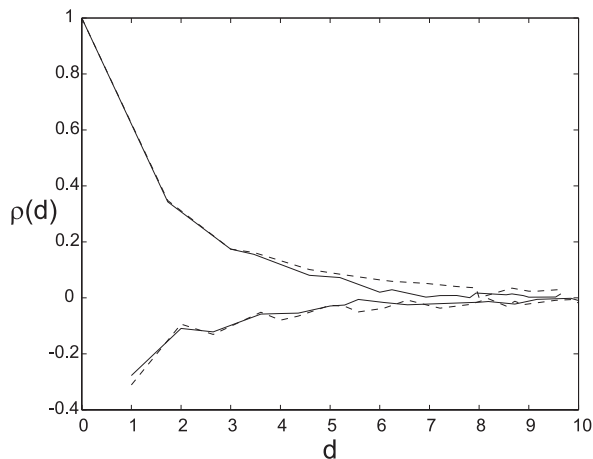


Figure 6: Correlation coefficient versus separation for the micrograph shown in Figure 2b (solid line) and the simulation (broken line) at $T = 0.55$. The upper curves are for sites on the same sublattice and the lower curves for sites on different sublattices.

6 Conclusions

The myosin lattice of some vertebrate muscles exhibits an interesting form of disorder in which the

orientations of the myosin filaments adopt one of two orientations that are distributed in a statistical fashion with short-range order. The characteristics of the disorder suggest a classical, frustrated, antiferromagnetic Ising model. Comparison of the second order statistics derived from micrographs with those from Monte Carlo simulations of the antiferromagnetic triangular Ising model support this hypothesis with an effective temperature in the range of $0.45 < T < 0.65$. These results will have implications for muscle structure and the simulation and analysis of x-ray diffraction from muscle specimens.

7 Acknowledgements

We are grateful to the N.Z. Marsden Fund for financial support, and to John Squire and Pradeep Luther (Imperial College, London) for discussion and provision of the micrographs.

References

- [1] J. M. Squire, "The structural basis of muscular contraction," London.: Plenum Press, 1981.
- [2] J. M. Squire, "Architecture and function in the muscle sarcomere," *Curr. Opin. Struct. Biol.*, vol. 7, pp. 247–257, 1997.
- [3] P. K. Luther and J. M. Squire, "Three-dimensional structure of the vertebrate muscle A-band II. The myosin filament superlattice," *J. Mol. Biol.*, vol. 141, pp. 409–439, 1980.
- [4] R. P. Millane and A. Goyal, "Analysis of the disordered myosin lattice in muscle," *Fibre Diffraction Review*, vol. 4, pp. 6–1, 2000.
- [5] B. Bödvarsson, S. Klim, S. Mortensen, M. Mørkebjerg, J. Chen, J. R. Maclaren, C. H. Yoon, P. K. Luther, J. M. Squire, A. Bainbridge-Smith, P. J. Bones, and R. P. Millane, "Determination of myosin filament positions and orientations in electron micrographs of muscle cross-sections," in *Image Reconstruction from Incomplete Data III* (P. J. Bones,

- M. A. Fiddy, and R. P. Millane, eds.), vol. 5562 of *Proc. SPIE*, pp. 97–108, 2004.
- [6] C. H. Yoon, J. Chen, J. R. Maclaren, B. Bödvarsson, S. Klim, S. Mortensen, M. Mørkebjerg, P. K. Luther, J. M. Squire, A. Bainbridge-Smith, P. J. Bones, and R. P. Millane, “Automated analysis of electron micrographs of muscle cross-sections,” in *Proc. Image and Vision Computing New Zealand 2004*, pp. 173–179, 2004.
- [7] G. H. Wannier, “Antiferromagnetism. The Triangular Ising Net,” *Physical Review*, vol. 79, p. 357, 1950.
- [8] R. Moessner and S. L. Sondhi, “Ising models of quantum frustration,” *Physical Review B*, vol. 63, p. 224401, 2001.
- [9] M. E. J. Newman and G. T. Barkema, *Monte Carlo Methods in Statistical Physics*. Oxford, UK: Oxford University Press, 1999.

Vision based Human Activity Detection for Eldercare and Security

Nigel Pereira, Liyanage C. De Silva, and Amal Punchihewa

Institute of Information, Sciences & Technology, Massey University, Palmerston North, New Zealand

Email: nigel.ep@gmail.com, L.desilva@massey.ac.nz, g.a.punchihewa@massey.ac.nz

Abstract

This paper presents a vision based human movement and activity detection system for home-care environments or home security applications. We define an event and spatial relationship based approach to the problem with state transition. Using our proposed techniques we detect and track people entering into a room and classify actions at 96.1% accuracy. Using the centre of gravity based tracking we detect falling of a person with high accuracy.

Keywords: video based event detection, home-care, security, background segmentation.

1 Introduction

Ambient Intelligence in Home-care applications is growing at a very fast pace in all parts of the world. One main requirement of such applications is the human detection and activity classification. The necessity for the development of human detection methods in the field of modern Home-care and security systems has become very popular and essential. There are many techniques currently being used for human detection. It is necessary to detect the presence of the human in advance before processing the human activities such as falling, standing or walking etc[1].

Human detection techniques at present can be either video based or any other sensor based. Sensor based detections are such as [2], [3] and [4] where infrared sensors and carbon dioxide sensors are used to detect motion and magnetic sensors are utilized to detect the opening and closing of doors. An illumination sensor is a type of sensor where once the subject is present, the sensor relies on changes in the environment caused by the subject to trigger a chain of events in the circuit. A more fascinating approach is a system called Cenwits [5] Connection-less Sensor-Based Tracking Using Witnesses. This is a mobile system that emits a signal from time to time using RF communication. When two of these mobile sensors are close to each other, information is gathered such as time and location at that time of the subject carrying the sensor and finally all information is dumped at an access point. This system would be useful for application in a large area where it being necessary to keep track of individuals.

A camera based detection approach is given in [7] where it involves a single camera tracking a number

of people. The system works by extracting points and identifying feature points from an image, creates a path and clusters them and finally each of these clusters corresponds to a person. The W⁴: Who? When? Where? What? [8] technique relies on the system to solely identify a combination of shapes and sizes from the image segmentation of the monochromatic imagery to identify a subject's presence and its interaction and time. The system in [9] uses multiple cameras to detect human motion by selecting the best viewpoints of the images to extract a maximum amount of information on the individual or multiple amounts of individuals. The results of the system are reconstructions of the human position, normal axis and body size.

Our proposed approach involves video pre processing to obtain human activities in a room using single or multiple cameras. The aim of this project is to design a system that can detect the activities being carried out in a room by one or several individuals. Essentially, we would want to be able to determine if there is anyone present in the room, if there is then we would like to determine what that person is doing in the room.

2 Approach

2.1 Video Based Activity Detection

Many types of vision-based systems for surveillance and monitoring of closed environments have been described and built over the past 20 years [13]. Henry Tan et. al. [14][15] has proposed a simple technique for human activity recognition using head movement detection. *Smart environments* are an immediate application of human activity detection. Alex Pentland's research group at MIT Media Laboratory

designed a *smart room* in 1991 [16]. This has evolved from its initial design to its current state of five networked smart rooms in the United States, Japan and the United Kingdom. These rooms use several machines, none more powerful than a personal computer, to identify the location, identity, facial expression and hand gestures of the persons in the room. Few more related research can be found in [17][18].

Here we propose a system that analyses image sequences from multiple stationary cameras, acquired from a particular scene, to detect humans and their actions, and index the sequence according to these activities. The image sequences are indexed using the results for faster searching. *Key frames* are extracted from the image sequences for each entry in the index, to facilitate visual inspection without browsing the image sequence. In addition to the index, visualizations of motion paths for humans in the scene are created to provide a faster way of tracking human movements in the scene.

2.2 Background Initialization

An outline of the background initialization phase is shown in Figure 1

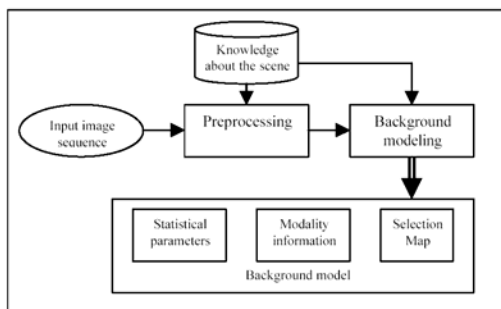


Figure 1 Background Initialization

2.3 Video segmentation

Initially we used 10 minutes long video, which then divided into several smaller segments. Each segment displays different activities being carried out in the room. These activities include a person walking, a person falling, and two people throwing objects to each other.

The reason for segmenting the video sequence is simply so that it can be processed by MATLAB easily. Segmenting the video will reduce the processing time done in MATLAB. After breaking these segments down, we can then analyze each activity in the room, and try to extract the characteristics of each activity carried out in the room. These characteristics can then be used to determine the activity carried out in the room, for successful human activity recognition in a real time situation. Figure 2 shows an activity detection scenario.

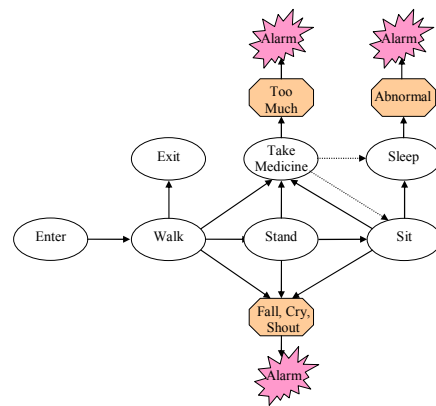


Figure 2 Activity Detection and Alarm Generation Scenario

3 Human detection

This part of the project will involve the detection of a person in the room, or the detection of a displaced object. In each frame after processing, the region of interest will be marked around the person in motion, or the displaced object.

3.1 Background analysis

The technique we used here to detect a human in the room or a displaced object was a background subtraction technique. When analyzing the background with no disturbance, we find that over several frames, the value for a particular pixel will change. These values when modelled by a histogram produce a bell shaped curve, indicating that the variations in a pixel that is a part of the background is normally distributed.

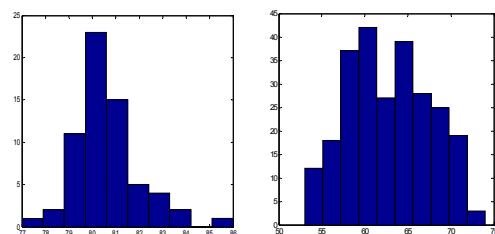


Figure 3 Histogram of 2 random pixels taken in the background.

In order to eliminate background noise (i.e. the variations in the background pixels), we get the frames, and obtain the minimum and maximum values for each pixel and store these values into 2 frames; one for the minimum pixels, and the other for the maximum pixels.

Using this information on the background, we can remove the background from the frames. Each frame will be compared to the minimum and maximum background frames. If a particular pixel lies outside its background range based on that pixel's minimum and maximum value, we can conclude that there is a significant change in the background there.

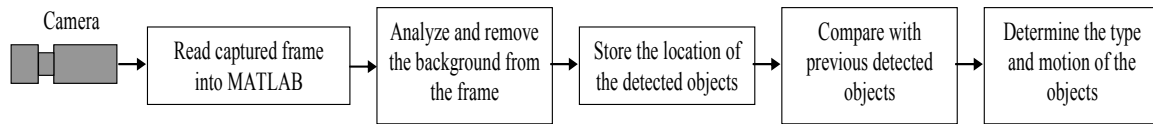


Figure 4 Object Detection Scenario

3.2 Frame division

Although we have blocked out pixel noise using the background subtraction technique, there is still some noise present in the frames, which could not be removed using the background subtraction technique.

To avoid these noise pixels, we divided the frame into smaller sections. We then check each section for the number of pixels that are not part of the background using the frame subtraction technique. If more than 30% of pixels in a section are determined to be not part of the background, we can say that there is a significant presence in that background.



Figure 5 Frame of subdivided background

4 Activity recognition

After determining the region of interest around the person or displaced object, we can obtain the centre point of the region. Based on the centre point of the region, we can assess the motion of the object (i.e. we can determine if the object is moving to the left, moving to the right, stationary, or falling).

If there is some degree of separation between objects or persons in a room, the program will be able to tell that there are two objects/persons present in the room, and will be able to analyze the centre points of these regions of interests separately.

5 Proposed Algorithms

Using technique discussed in section 3.2, we find that by subdividing the frames into smaller sections, the noise in the background has been reduced significantly. Also, the region of interest is now more concentrated around the person walking.

We will now look at the algorithms used for each technique. First we look at the background analysis and then background subtraction techniques.

5.1 Background analysis

This technique compares the frames that show the background of the room to get the minimum and maximum values for each pixel. The algorithm for this technique is shown in Figure A-1.

5.2 Background subtraction technique

This technique uses the information obtained from the background analysis section to check each frame. If a frame has any pixels that lie outside the range given by the minimum and maximum background values for that particular pixel, then we can conclude that there is some motion present there. The algorithm for this technique is shown in Figure A-2.

5.3 Frame division technique

This technique divides the frame into smaller sections. We then check to see how many pixels in each section lie outside the background range. If the number of pixels that are not a part of the background exceeds a particular threshold, then we can say that there is significant motion in that section. The algorithm for this technique is shown in Figure A-3.

6 Results

Some results are shown in Figure 6. Here we can see a set of frames to be processed by MATLAB, showing a person walking in the room. We first performed the background subtraction technique as described in section 3.1. As can be seen, the person is marked with a square around him. Also, the pixels that show the person were manipulated to show that the person is being picked up by the program.



Figure 6 (a) 3 original frames of a person walking (b) detected person using the Background Subtraction algorithm (c) detected person using the previous algorithm with the Frame Division Algorithm

6.1 Evaluation of Event Recognition

Several image sequences containing different actions and events were used to evaluate the accuracy of action and event recognition. Table 1 shows the accuracy of recognition of events in our proposed system.

Table 1 Accuracy of Action and Event Recognition

Actual action/ event	Classified correctly	Classified incorrectly	Not classified	Accuracy %
Enter (10)	10	0	0	100
Walk (27)	25	2	0	92.6
Exit (10)	10	0	0	100
Stand (14)	12	0	2	85.7
Sit (13)	12	0	1	92.3
Use PC (18)	17	1	0	94.4
Take object (10)	10	0	0	100
Place object (10)	10	0	0	100
Unusual event (10)	10	0	0	100
Overall avg. accuracy				96.1%

6.2 Fall Detection

Falling is one of the key events that we have investigated in our Home-care application since a sudden fall can lead the person into life threatening situations. A sample result sequence is shown below. Our foreground background segmentation algorithm clearly identifies the person who is falling (indicated in blue pixels) and triggers an alarm. The falling detection is identified by calculating the centre of gravity of the large blob that is formed by the person. When the person is falling, the centre of gravity goes

down steeply. The percentage accuracy of falling verses walking detection was very high for all the video sequences we have investigated.



Figure 7 Fall Detection

7 Conclusions

Using our proposed techniques we were able to detect and track people entering into a room and classify actions at 96.1% accuracy. Using the centre of gravity based tracking we were able to detect falling of a person with high accuracy. In this paper we have focused only on the video based activity recognition but we are developing a multimodal activity detection scheme in which audio and other sensor integration can improve the accuracy of the activity recognition and also reduce the privacy problem inherent to video based detection.

References

- [1] Jongwoo Lim; D Kriegman, "Tracking humans using prior and learned representations of shape and appearance", published in the proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. FG 2004, 17-19 May 2004, pp. 869-874.
- [2] Ogawa, M., Togawa, T., "Monitoring Daily Activities and Behaviours at Home by Using Brief Sensors", *1st Annual International IEEE-EMBS Special Topic Conference on Microtechnologies in Medicine & Biology*, October 12-14, 2000, Lyon, France, pp 611-614.
- [3] Ogawa, M., Ochiai, S., Otsuka, K., Togawa, T., "Remote Monitoring of Daily Activities and Behaviors at Home", *2001 Proceedings of the 23rd Annual EMBS International Conference*, October 25-28, Istanbul, Turkey, pp 3973 – 3976 Vol. 4.
- [4] Ogawa, M., Suzuki, R., Otake, S., Izutsu, T., Iwaya, T., Togawa, T., "Long-Term Remote Behavioural Monitoring of the Elderly using Sensors Installed in Domestic Houses", *Proceedings of the 2nd Joint EMBS/BMES Conference*, October 23-26, 2002, TX, USA, pp 1853 – 1854, Vol. 3.
- [5] Huang, J. H., Mishra, S., "A Sensor based Tracking Syst. using Witnesses", *25th IEEE Int. Conf. on Distributed Computing System Workshops*, 2005, pp 251-255.
- [6] Kainka, B., "Passive-Optical Person Detector", http://www.kfupm.edu.sa/club/ieee/Projects/pr_oj007.pdf, visited on 02/12/2004.
- [7] Segen, J., Pingali, S., "A Camera-Based System for Tracking People in Real Time", *Proceedings of the 13th International Conference on Pattern Recognition*, 1996, pp 63 – 67, Vol. 3.
- [8] Haritaoglu, I., Harwood, D., Davis, L. S., "W4: Who? When? Where? What? A Real Time System for Detecting and Tracking People",

- Proceedings of the 3rd IEEE International Conference on Automatic Face and Gesture Recognition*, April 14-16, 1998, pp 222 – 227.
- [9] Utsumi, A., Mori, H., Ohya, J., Yachida, M., “Multiple-Human Tracking using Multiple Cameras”, *Proceedings of the 3rd IEEE International Conference on Automatic Face and Gesture Recognition*, April 14-16, 1998, pp 498-503.
- [10] Liyanage C De Silva, and Tsutomu Miyasato, “Hierarchical Expression Recognition”, Japanese National Patent – No 2967058, awarded in August 1998.
- [11] Liyanage C De Silva, Tsutomu Miyasato, and Ryohei Nakatsu, “Facial Emotion Recognition Using Multimodal Information”, in proceedings of IEEE Int'l Conf. on Information, Communications and Signal Processing (ICICS1997), pp. 397-401, Vol. 1, Singapore, Sep. 1997.
- [12] Liyanage C De Silva, “Audiovisual emotion recognition”, *Invited paper* in the proceedings of IEEE International Conf. on Systems, Man and Cybernetics (SMC2004), The Hague, The Netherlands, Oct. 10-13, 2004.
- [13] A. Pentland, T. Choudhury, “Face Recognition for Smart Environments”, *Computer*, pp. 50-55, IEEE Press, United Kingdom, February 2000.
- [14] Henry CC Tan and Liyanage C De Silva, “Human Activity Recognition by Head Movement using Elman Network and Neuro-Markovian Hybrids”, in proc. of Image and Vision Computing New Zealand 2003 (IVCNZ 2003), pp. 320-326, Massey Univ., Palmerston North, New Zealand, Nov. 26-28, 2003.
- [15] Henry CC Tan, E G Ruwan Janapriya, and Liyanage C De Silva, “An Automatic System for Multiple Human Tracking and Action Recog. in an Office Environment”, in proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2003), Hong Kong, IMSP-L2.6, April 6-10, 2003.
- [16] A. Pentland, “Smart Rooms”, (<http://vismod.www.media.mit.edu/vismod/demos/smartroom/ive.html>).
- [17] G. C. de Silva, “Traffic Flow Measurement Using Video image Sequences”, M. Eng. Thesis, Department of Computer Science and Engineering, Univ. of Moratuwa, Sri Lanka, 2001.
- [18] A. Utsumi, H. Mori, J. Ohya, M. Yachide, “Multiple View-Based Tracking of multiple Humans”, in proc. of the 14th Int. Conf on Pattern Recognition, 1998, Vol.1, pp.597 -601, 1998.

Appendix:

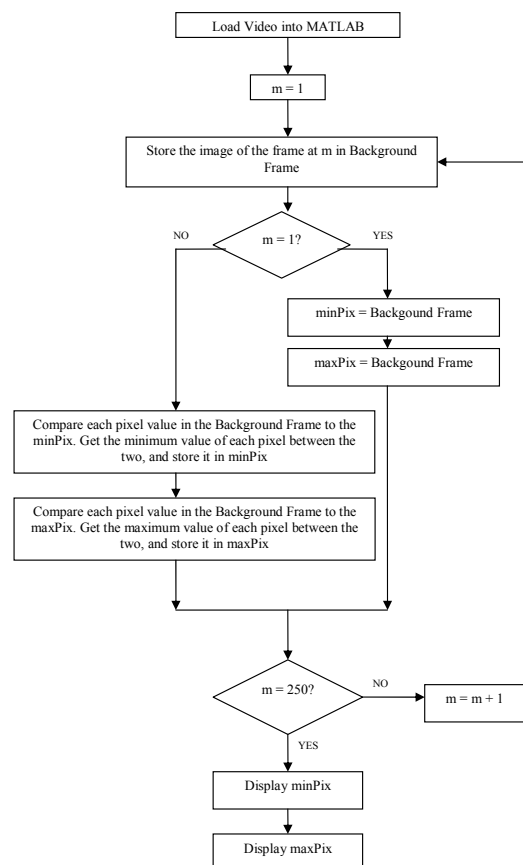


Figure A-1 Background analysis

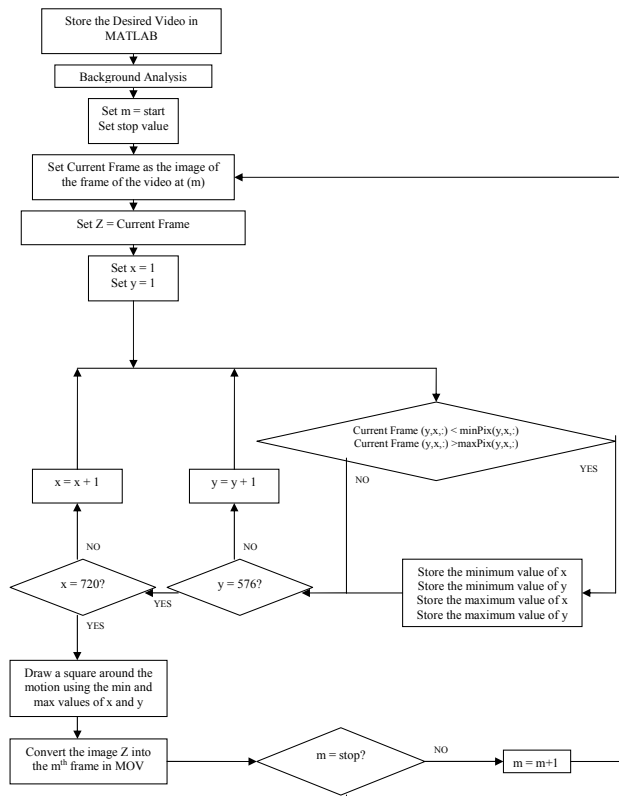


Figure A-2 Background subtraction technique

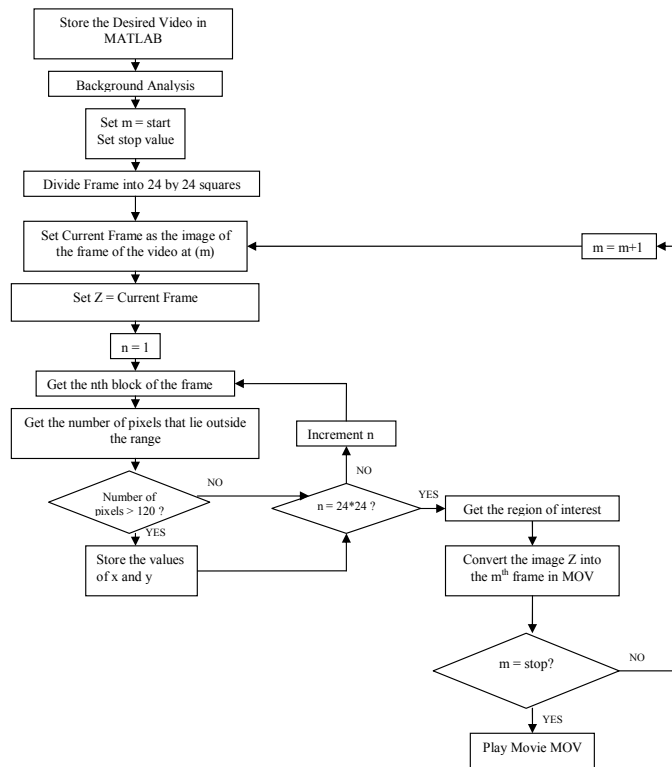


Figure A-3 Background subtraction and frame division technique

Automatic Recognition of Light-Microscope Pollen Images.

G.P.Allen¹, R.M.Hodgson¹, S.R.Marsland¹, G.Arnold¹, R.C.Flemmer², J.Flenley³,
D.W.Fountain⁴

¹ Massey University, Institute of Information Sciences and Technology.

² Massey University, Institute of Technology and Engineering.

³ Massey University, Geography Programme, School of People, Environment and Planning.

⁴ Massey University, Institute of Molecular BioSciences.

Email: g.p.allen@massey.ac.nz

Abstract

This paper is a progress report on a project aimed at the realization of a low-cost, automatic, trainable system “AutoStage” for recognition and counting of pollen. Previous work on image feature selection and classification has been extended by design and integration of an XY stage to allow slides to be scanned, an auto-focus system, and segmentation software. The results of a series of classification tests are reported, and verified by comparison with classification performance by expert palynologists. A number of technical issues are addressed, including pollen slide preparation and slide sampling protocols.

Keywords: pollen recognition, image processing, classification, microscopy.

1 Introduction

Fossil pollen analysis is used to determine flora genus from which climate data, evidence of human activity and oil deposit locations, can be deduced. Honey type, and location of origin, can be indicated by the pollens found in the honey. Allergy sufferers can be advised of high pollen counts in the air. Forensic investigations can be aided by determining if an object has been in a certain general location by identifying the pollen types attached.

The need for an automated pollen counting system has been identified and detailed for many years [1]. A previous paper reported on progress toward such a system [2] and a significant milestone in that project is reached, and reported here, with the complete system designed, built and evaluated as a functioning unit.

The system will:

- reduce the massive amount of laborious counting required by highly skilled people involved in palynological endeavours (30 months in a PhD);
- increase sample quantities allowing more accurate pollen studies, especially in fine resolution sampling [3];
- increase the frequency and locations of pollen counts, which are of use to inhalant allergy and asthma sufferers.

A good description of the problems involved and requirements of a complete automated system have been described recently [4, 5]. The broad requirements are to locate pollens on a microscope

slide and classify each into taxonomic categories at reasonable cost, and with a success rate at least that of a skilled person. The saving is labour, and time consumed by people with skills that could be better applied to less mundane tasks.

The steps involved in the AutoStage project are:

1. develop a set of features derived from optical images of pollen that are discriminable. [6]
2. develop a supervised classification system based on the features-set developed in step 1.
3. design a suitable low cost digital microscope [7]
4. develop an image segmentation scheme to isolate images of pollen and exclude detritus
5. develop and build an XY stage to allow slides to be scanned using transmitted or reflected light
6. develop a system to find the location of pollen on a slide and to capture in-focus images
7. integrate the system resulting from steps 1-6
8. evaluate and verify classification and count performance of the system, and compare to trained palynologists.

Steps 1-3 were completed [2]. This project is to develop and build a working microscope, build in an XY stage and focus hardware, develop working segmentation and focus algorithms: steps 4-8. We report development of the final stages and describe the completed system that takes a prepared slide and captures microscopic images from which



Figure 1: AutoStage

pollen are segmented, image features extracted and pollen taxa classified and counted.

2 Automated System Description

The system described here finds pollen grains on a slide and captures images of them together with their location information. Image features are extracted and used for classification of pollen types, enabling a count of the number of grains of each pollen type. The classification of pollen can be manually checked.

Selection of any portion of a slide to be processed is accomplished by the user moving the camera to opposite corners of a rectangular area of interest. The current system is capable of capturing areas shaped with a pixel resolution of $1/2$ micron.

The system comprises:

1. a machine to capture the images (§2.1)
2. segmentation, auto-focus and classification algorithms (§2.2)
3. a computer to run the algorithms and control the hardware (§2.3)

In addition to the sub-systems, slide preparation (§2.4) and slide sampling (§2.5) are discussed.

2.1 The Machine

The ‘machine’, is an XY stage with attached slide holder. Two digital microscopes are solidly mounted above a filtered and cooled light source. As transmission lighting is used, the slide sits on an aperture in the XY stage positioned between the cameras and light source as in Figure 2.

There are two power supplies for lighting and stepper motors. Two motors move the XY stage to locate pollen under the microscope and a third motor adjusts the relative height of the cameras for focussing.

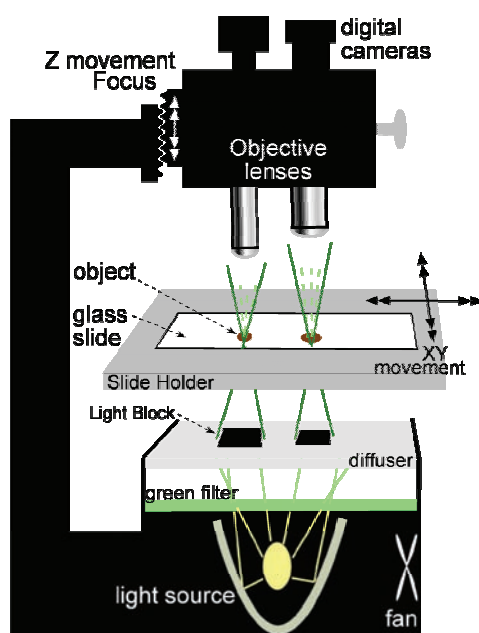


Figure 2: AutoStage elements

2.1.1 The Stage

The slide is held in a standard microscope holder and is moved by a commercial XY precision stage driven by two stepper motors. The motors are micro-stepped to $1/10^{\text{th}}$ of their 1.8° step angle, allowing a linear movement of 2.6 microns per step (the smallest pollen of interest is about 10 microns across). The field of view of the high magnification camera is 165×123 steps. The speed of movement is set below maximum to about 5mm per second.

2.1.2 Two Microscopes

A low magnification microscope with a large field of view (FOV), locates pollen grains quickly while a high magnification microscope captures images with sufficient detail for feature extraction.

A digital camera sensor and a standard microscope objective lens placed 207mm from the camera sensor plane, forms the ‘high magnification’ microscope with an optical magnification of $11.2\times$. Because the camera sensor elements are 4.65 microns square, the magnification that is required for a human to view the formed image occurs in translation from a 1024×768 pixels in the 6mm diagonal rectangle of the sensor, to 1024×768 pixels on a computer screen. That is about $72\times$, and $720\times$ including optical magnification.

The small *optical* magnification results in a depth of field greater than for a conventional microscope with the same overall magnification.

The FOV of the main camera is less than half a millimetre square. To image an entire slide more quickly, the low magnification camera with about $1/10^{\text{th}}$ the magnification, is used to more quickly cover the slide and locate potential pollen grains. A segmentation algorithm identifies most detritus and the locations of remaining objects found are stored for the high magnification camera to investigate.

Segmentation, using the high magnification camera and finding an acceptable object, produces an image slightly larger than the object bounding rectangle. The image is stored for feature extraction and classification (Figure 3).

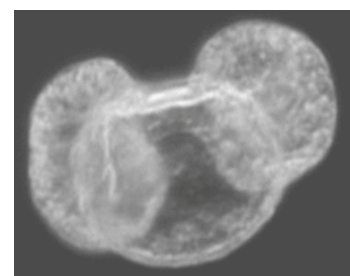


Figure 3: hi-mag segmented image (*Pinus radiata*. $\sim 50\mu\text{m}$)

2.1.3 The Lighting

Lighting is provided by a simple arrangement of a quartz halogen lamp directly below the cameras, with filtering, and a fan for cooling. One filter is a band-pass to reduce any chromatic aberrations caused by the objective lens. A green filter was chosen

because the camera is filtered to have a maximum sensitivity in the same area of the spectrum as human vision, $\lambda \approx 550\text{nm}$: green.

A diffusion filter is the topmost filter and has a light blocking rectangle below each camera. The diffused light therefore strikes the object oblique to the optical axis, making it a simple form of “dark field” illumination. Little of the light direct from the source enters the objective lens directly so the background is dark and objects are light with darker ‘shadows’ formed by the surface features. Contrast is increased over light-field transmission microscopy with one study measuring an increase from 10% to 85% contrast [8]. Sub-resolution visualisation is another property of dark-field illumination [9]. This is where objects smaller than the resolution of the optical system are indicated, but not resolved. That this has a positive or negative effect on image features extracted in this case would require further study.

The dark-field effects are helpful for finding pollen in the low magnification camera and creating a better image for feature extraction.

2.2 The Algorithms

2.2.1 Auto-Focus

The low magnification camera is initially focussed manually at the same time the user is setting the limits for a region of interest within the total area of the slide. The auto-focus software then steps the camera through that manually set focus position, to refocus. The auto-focus operates by calculating the standard deviation of all grey levels of each image as it steps through the focal plane. The sequential values are stored as a vector and a suitable peak is located by a “local maximum” algorithm. The camera is moved back to the step where the local maximum was found. Movements of critical placement are always in the upward direction. This focus position is then used for all images taken with the low magnification camera as a high depth of field keeps pollen sufficiently in focus. There are several focus measurement methods in the literature [10-13]. After experimentation, the standard deviation function was chosen for the low magnification microscope as it has a desired smoothing effect and it is not computationally demanding.

The high magnification camera is fixed on the same focus movement so once the low magnification camera is focussed, the high magnification camera can be moved to a near focus position. This position is used to perform an automatic refocus. Auto-focusing is performed on each object because the pollen grains are not necessarily all within the same focal plane and depth of field is less for this microscope.



Figure 4: glass slide with cover slip

The auto-focussing algorithm used with the high magnification camera incorporates a squared gradient measure where for each pixel, the maximum grey-scale gradient-squared, between y direction and x direction is chosen and all chosen values summed.

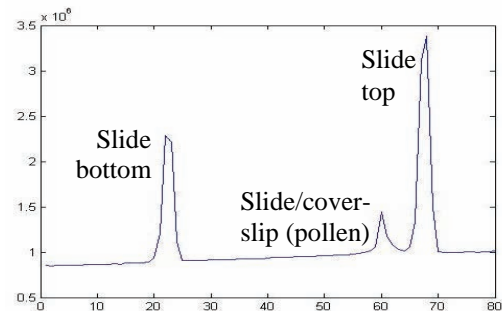


Figure 5: plot of focus image against gradient with a dirty slide giving greater focus values at the outer surfaces. Centre peak is the focus aim.

The values plotted against focus step number, results in a large ‘spike’ in value for 3 or 4 steps of the focus movement. To improve the auto-focus, the step size would need to be made smaller and an algorithm with greater selectivity might then be used. To reduce computation time and help ensure the object of interest is in focus, the image area is reduced to around the centre of the image where the object may be located.

It takes 15s for one complete pollen grain capture: move stage; auto-focus; capture; segmentation, save image. Auto-focus takes $\frac{2}{3}$ of that time at 10s.

2.2.2 Segmentation

Segmentation is difficult and often problem specific. For a review on segmentation techniques see [14].

A stored background image, taken with no slide in place, is subtracted from images captured to remove any image anomalies caused by the system. Objects are located by first finding edges using a Sobel edge operator. As pollen are small objects with well defined outlines, then the edge detection results in a mostly closed loop. Morphological operations follow: dilation, to join any broken edges; filling any closed loops to form solid ‘blobs’. Erosion then reduces the blob size to be close to that of the original object.

The blob pixel counts are measured, and any blobs too small or too large to be a pollen grain are removed. The smallest pollen grains of interest (about 10 microns across) have a blob area of 5 pixels in an image from the low magnification camera. Large pollen grains, 100 microns across, are represented by a blob area of about 500 pixels.

For each blob of correct size, a bounding rectangle and its area are calculated. If the rectangle has an aspect ratio too small, or the blob area to rectangle area ratio is too small, then the blob is removed.

The area of a convex hull for each blob is calculated and if the blob area to hull area ratio is too small, the object is removed.

The centres of remaining blobs are found and their positions on the slide calculated and stored. The high magnification camera is moved to each of those positions and performs a segmentation process to find a valid object nearest the centre of the image. Tolerances in movements cause the object to appear with a variable offset.

2.2.3 Classification

To perform taxonomic classification, image features extraction and a multi-layer perceptron [15] are used in line with [16]. The features used are those identified in [17] consisting of 43 shape and texture features.

Texture features are represented by a series of Wavelet transforms that measure localised spatial/spatial-frequency content using Gabor and Orthogonal Wavelet transforms. Orientation sensitivity is reduced by averaging the results corresponding to different directions [6]. Other textural features used are Grey Level Co-occurrence Matrix, and Grey Gradient Co-occurrence Matrix. Shape features are geometric, histogram and second moment.

Linear Discriminant Analysis, together with Principal Components Analysis, were employed to compare discrimination and check for any redundant features [18]. No reduction of feature-set size was found useful. A Support Vector Machine algorithm, with its binary classification capability, was used to discriminate two grass pollens and found to be less effective than the multi-layer Perceptron.

2.3 The Computer

The computer used is a PC with a 2.6GHz processor and 1Gbytes of RAM running Windows XP professional. All the code is written in Matlab including: image acquisition via USB and IEEE1394 (FireWire); control of the stepper motors via a serial port; and the auto-focus, segmentation, and classification algorithms.

2.4 Slide Preparation

To improve the efficacy of the system the slides should be prepared in a prescribed and suitable manner. It is important this should be similar to current practice.

Auto-focus can be adversely affected by objects on surfaces other than the top of the slide and the bottom

of the cover-slip. The segmentation algorithms could be compromised and images captured would be degraded if dust or oil were present, even if they were out-of-focus.

The prescription proposed is for the pollen samples to be suspended in some setting gel. Silicon oil is suitable and may be desirable if the slides are to be checked on a conventional microscope, as are agar or glycerol if an aqueous medium is required. The suspension should have a concentration that results in no more than 500 pollen grains per slide to reduce clumping. The sample medium volume and viscosity is such that when dropped onto the slide and the cover slip is placed on top, the medium does not travel past the outer edges of the cover slip.

The slide is placed on a warmer to allow air bubbles to escape the gel. Wax is dropped onto the slide at the edge of the cover slip to 'wick' under the cover slip to seal the pollen suspension in, and hold the cover slip firmly in place. The slide surfaces can now be cleaned without moving the pollen grains within the slide. Adding detergent to a last rinse will help reduce clumping.

2.5 Spatial Sampling of Slides

If sampling the slide is applicable, the high magnification camera only might be utilised. It may perform sampling better than in the current methods of manual counting.

It is proposed that the area of interest of the slide be divided up into rectangles, a sample of those rectangles randomly selected, and that the camera capture an image of each selected rectangle. The images would be segmented, classified and counted for each rectangular sample. A statistical analysis would estimate the slide populations of each pollen type.

By running trials on slides with known populations, a suitable sample size could be calculated.

This should prove a better method than the present manual methods, as the randomness of the present slide sampling approach is suspect [19].

3 Experiments and Results

Three image data bases were compiled:

1. CM: captured using a conventional microscope
2. AS: captured using AutoStage
3. BR: images used by France et al. [4]

A selection of the data base images was made of 50% for training, 25% for validation and 25% for the final tests reported here. The validation set was used with the training set to adjust neural net parameters for optimum results and verify the system working. The training and validation sets were then combined for training and the test set used for the final test. The feature sets extracted from the images, were presented

in random order to the classification software. Results are expressed as total correctly classified pollens as a percentage of all pollens, and the means and standard deviations over 5 tests recorded.

3.1 Compare AS with CM

The aim of this experiment is to compare classification results using images taken from the same slides by AutoStage and by a conventional microscope.

Test description: Take 40 training, 10 test and 7 types of images from AS and CM data bases. Classify both sets and compare mean results and check for difference with a Students t test.

Results: The AS mean was **98%** correct (sd = 1.2) and the CM mean was **94%** correct (sd = 0.6). Using a 95% confidence t-test, the means are significantly different.

3.2 Classification of Grass Pollens

The aim of this experiment is to check performance of the AutoStage when classifying grass pollens which are commonly counted as one type as they are very difficult to distinguish manually under a light microscope.

Test description: take 3 grass pollen image sets from the AS data base, using 150 training and 50 test images. Classify the sets.

Results: Mean = **90%** correct (sd = 0.3).

3.3 Large Pollen Type Count

The aim of this experiment is to check the performance of the AutoStage using a wider range of pollen types in a single test.

Test description: 19 types were used for the experiment including all types available, however 2 of the 3 grass pollens were excluded. 150 training and 50 test images were used.

Results: Mean = **89%** correct (sd = 0.5).

3.4 AS Compared With another Project

The aim of this experiment is to compare AS classification results, to results recorded by France et al [4].

Test description: France, recorded results using 3 pollen types with 60/60/84 images made available on the internet. Here, 45 of each set of these images were used for training and 15 images for testing. Validation was not done as the neural network configuration and weights were not altered from other tests.

Results: France achieved overall **82%** correctly identified in the final classification stage with 3%

being misclassified and 15% being rejected. The AS was, on average, **95%** successful in distinguishing 15 of the same images with 5% misclassification.

3.5 AS Compared with Experts

The aim of this experiment is to compare the total process of pollen counting from a slide by the AutoStage, with the count of the same slide by experts.

Test description: A slide with 6 pollen types is prepared. Five 'experts' including two professors, a post doctoral student, a technician working in palynology and an honours student, count the slide. The AutoStage then counts the slide.

Result. The table below shows statistics of the human count and one AutoStage count.

Pollen type	5 People			AutoStage
	Mean	StdDev	Range	Raw Count
1	65.6	13.4	43 - 77	64
2	14.2	4.8	9 - 20	13
3	21.8	8.7	16 - 37	18
4	86	17.9	58 - 102	75
5	0.8	0.4	0 - 1	1
6	8.6	1.5	7 - 11	7

Table 1: The performance of AutoStage was compared to five human experts.

4 Conclusions

1. Most importantly, for a *complete working system* and functional test described in §3.5, AutoStage has matched the result of experts. The *variability* of AutoStage has yet to be determined with multiple counts by AutoStage on more slides and a comprehensive statistical analysis.
2. The AutoStage system is giving classification results improved upon known published results.
3. The system is completed, functions well with promises of the ability to meet the requirements to be useful to a palynologist.
4. Images from the AutoStage used for classification performed better than images from a conventional microscope.
5. The lighting system described gives images of excellent contrast.
6. The auto-focus system performs well. The digital microscope, having a greater depth of field than a conventional microscope, makes focussing less critical.
7. The XY stage, with movement limits larger than a slide, a repeatability of position of 20 microns, speed in excess of 10mm per second, and a spatial resolution of 2.6 microns, would be satisfactory for a manufactured product.
8. The component costs of the prototype system were under \$NZ15,000 including the computer.

5 Acknowledgements

Many thanks to Steve Denby and crew at the mechanical workshop of the Institute of Fundamental Sciences, Massey University, who built the AutoStage. Thanks also to, Xiuying Zou for conventional image capture of reference pollens.

6 References

- [1] E. C. Stillman and J. R. Flenley, "The needs and prospects for automation in palynology," *Quaternary Science Reviews*, vol. 15, pp. 1-5, 1996.
- [2] R. M. Hodgson, C. A. Holdaway, Z. Yongping, D. W. Fountain, and J. R. Flenley, "Progress towards a system for the automatic recognition of pollen using light microscope images," *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*, pp. 76 - 81 2005
- [3] D. G. Green, "The Ecological Interpretation of Fine Resolution Pollen Records," *New Phytologist*, vol. 94, pp. 459-477, 1983.
- [4] I. France, A. W. G. Duller, G. A. T. Duller, and H. F. Lamb, "A new approach to automated pollen analysis," *Quaternary Science Reviews*, vol. 19, pp. 537-546, 2000.
- [5] M. Rodriguez-Damian, E. Cernadas, A. Formella, M. Fernandez-Delgado, and P. De Sa-Otero, "Automatic detection and classification of grains of pollen based on shape and texture," *IEEE Transactions on Systems Man and Cybernetics Part C-Applications and Reviews*, vol. 36, pp. 531-542, 2006.
- [6] Y. Zhang, D. W. Fountain, R. M. Hodgson, J. R. Flenley, and S. Gunetileke, "Towards automation of palynology 3: pollen pattern recognition using Gabor transforms and digital moments," *Journal of Quaternary Science*, vol. 19, pp. 763-768, 2004.
- [7] C. Holdaway, "Automation of Pollen Analysis using a Computer Microscope," vol. Masters. Palmerston North: Massey University, 2005, pp. 125.
- [8] P. C. Montgomery and J. P. Fillard, "Study of microdefects in near-surface and interior of III-V compound wafers by dark-field transmission microscopy," *Electronics Letters*, vol. 24, pp. 789-790, 1988.
- [9] A. I. Abdel-Fattah, M. S. El-Genk, and P. W. Reimus, "On Visualization of Sub-Micron Particles with Dark-Field Light Microscopy," *Journal of Colloid and Interface Science*, vol. 246, pp. 410-412, 2002.
- [10] F. C. A. Groen, I. T. Young, and G. Ligthart, "A comparison of different focus functions for use in autofocus algorithms," *Cytometry*, vol. 6, pp. 81-91, 1985.
- [11] N. Kehtarnavaz and H. J. Oh, "Development and real-time implementation of a rule-based auto-focus algorithm," *Real-Time Imaging*, vol. 9, pp. 197-203, 2003.
- [12] A. Santos, C. O. De Solorzano, J. J. Vaquero, J. M. Pena, N. Malpica, and F. Del Pozo, "Evaluation of autofocus functions in molecular cytogenetic analysis," *Journal of Microscopy-Oxford*, vol. 188, pp. 264-272, 1997.
- [13] J.-M. Geusebroek, F. Cornelissen, W. M. Arnold, and H. G. Smeulders, "Robust autofocusing in microscopy," *Cytometry*, vol. 39, pp. 1-9, 2000.
- [14] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern Recognition*, vol. 26, pp. 1277 1294, 1993.
- [15] I. T. Nabney, "NETLAB," Software, <http://www.ncrg.aston.ac.uk/netlab/index.php>, 2003.
- [16] P. Li and J. R. Flenley, "Pollen texture identification using neural networks," *Grana*, vol. 38, pp. 59-64, 1999.
- [17] Y. Zhang, "Pollen Discrimination Using Image Analysis," Massey University, Palmerston North report, 2001-2003 2003.
- [18] P. Etheridge, "Discrimination of Pollen Taxa from Digital Image Feature Data," Massey University, Palmerston North, Report July 28 2005.
- [19] D. Brookes and K. W. Thomas, "the distribution of pollen grains on microscope slides. Part 1. The non-randomness of the distribution," *pollen et spores*, vol. IX, pp. 621-629, 1967.

Tracking Articulated Objects using Improved Particle Filters

Martin Tosas, Li Bai

School of Computer Science and IT, University of Nottingham
Jubilee Campus, Nottingham, NG8 1BB, UK
mtb@cs.nott.ac.uk, bai@cs.nott.ac.uk

Abstract

This paper presents an application of particle filters and partition sampling to visual tracking of tree-structured articulated objects. The efficiency of particle tracker suffers as a result of particle propagation between time steps. This problem is resolved by using a novel technique, referred to as particle interpolation. An articulated hand contour tracking system is developed using the particle filters for digital entertainment applications and its performance is evaluated.

1 Introduction

Blake and Isard proposed a framework for object contour tracking using deformable templates and Condensation filters [1,4]. Advantages of this framework include robustness against cluttered backgrounds and efficiency of computation. The framework can be adapted to track articulated objects. However, the dimension of the configuration vector for many articulated objects, such as the human body or hand etc, is too large to be dealt with directly by a Condensation filter. MacCormick and Blake introduced a technique called *partition sampling* [2,4,5], which avoids the high cost of particle filters when tracking more than one object. Later, this technique was used by MacCormick and Isard [3] to implement an articulated hand tracker. They assigned a partition to each of the articulations in the hand, and treated these partitions as a chain even. However, it would be more efficient to take into account the tree structure of the hand. This paper addresses the use of partition sampling and a new technique *particle interpolation* for tracking tree-structured articulated objects.

The paper is organised as follows: section 2 presents an application of partition sampling to tracking tree-structured articulated objects, and describes the problems that will arise. Section 3 proposes a new technique, named particle interpolation, which resolves the problems discussed in Section 2. Section 4 presents a hand tracking implementation that uses these techniques. The

implementation is tested on two video sequences, and its performance is discussed. Section 5 concludes the paper.

2 Tracking Articulated Objects using Partition Sampling

An example tree-structured articulated object is shown in Figure 1(a) via a hand contour model. It assumes that the palm is always parallel to the camera's image plane, and independent finger and thumb movements are allowed. The hand contour model consists of a hand palm and five fingers, named L, R, M, I, T, representing Little, Ring, Middle, Index, and Thumb. Each finger can rotate around its pivot (the black dot at the base of the finger) to represent abduction/adduction movements. The length of the fingers can change to represent the 2D projection of a finger's flexion/extension movement. The angle and length of a finger are represented as α , and L respectively with the name of the finger as a subscript. The whole hand is allowed to translate by (x, y) , rotate around the hand palm pivot r , and scale by s . The articulated hand contour thus has 14 parameters.

The first step to use partition sampling with this model is to choose a convenient decomposition of the configuration space, for example: one partition for the hand palm, parameters (x, y, r, s) ; and one partition for each of the five fingers, each with parameters (α, L) . The next step is to define 6 particle sets, 6 motion models, and 6 measurement functions, related to the 6 partitions.

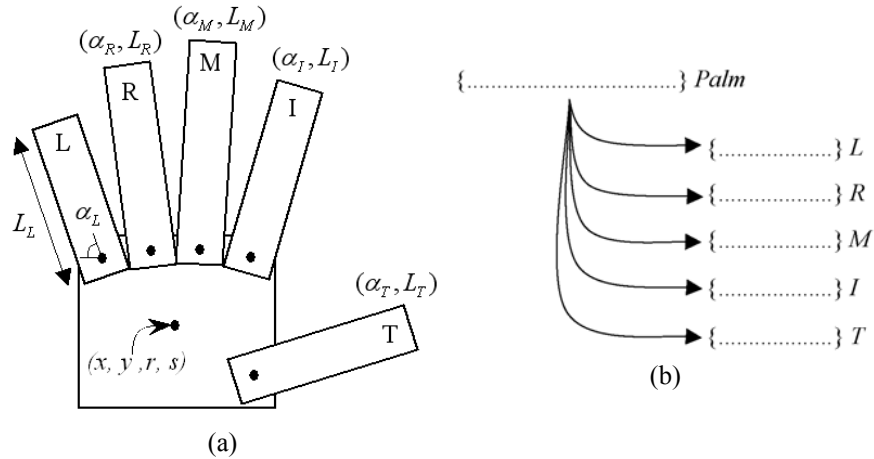


Figure 1: (a) Simplified articulated hand contour model with 14 DOF. (b) Particle set tree for the articulated hand model; *Palm* is the parent particle set, *L*, *R*, *M*, *I* and *T* are the child particle sets.

We define the hand palm partition as the parent partition, and the 5 finger partitions as the child partitions. The 6 partitions thus form a two-level tree, with 6 particle sets associated with each partition. The particle set tree is shown in Figure 1(b). Each of the particles in the parent particle set is connected to a particle in each of the child particle sets, so that each parent particle can access the child particle state, and vice versa. We refer to each of these parent/child particle groups as a *complete particle*. A tracking sequence for the hand model of Figure 1(a) starts with a single complete particle set whose parameters are aligned with the features of a hand in a video sequence.

The particle set tree is then processed following the algorithm in Figure 2. The parent particle set is referred to as *Palm* (being the hand palm particle set), and the child particle sets are referred to as *L*, *R*, *M*, *I*, and *T* respectively. In Step 1.1, each of the complete particles from the previous time step is used to generate a number of new particles in the *Palm* particle set, proportional to the weight of the complete

particle, and in relation to the total size of the *Palm* particle set. Steps 1.2, 1.3, and 1.4 constitute a Condensation time step on the *Palm* particle set, i.e. applying dynamics to the particles, weighting the particles, and resampling the particles. However, the resampling is different from a standard Condensation resampling in the sense that only the particles with the highest weight in the set are selected. In Step 2.1, the selected particles are used to generate new particles in the child particle sets. The number of new particles is proportional to the weight of the selected parent particle, and the total size of the child particle set. In Step 2.2, dynamics are applied to each particle in the child particle sets. The dynamics needs two previous states in order to predict a new state. In Step 2.3, the particles in the child particle sets are weighted. The weighting function is specific for each child partition. Step 2.4 selects child particles with the highest weight. Finally, in Step 3, complete particles are formed by grouping the selected particles in both the parent and the child particle sets.

1. For the parent particle set (*Palm*) do:
 - 1.1. Use the complete particles of the previous time step to generate new particles for *Palm*.
 - 1.2. Apply dynamics to each of the particles in *Palm*.
 - 1.3. Weight particles in *Palm*.
 - 1.4. Select particles from *Palm* that constitute peaks of weight in the set.
2. For each of the child particle sets (*L*, *R*, *M*, *I*, and *T*) do:
 - 2.1. For each of the selected particles in *Palm*, generate a number of new particles in the child particle set, proportional to the weight of the selected particle in *Palm*.
 - 2.2. Apply dynamics to each of the particles in the child particle set.
 - 2.3. Weight particles in the child particle set.
 - 2.4. Select particles, from the child particle set, that constitute peaks of weight in the set.
3. Form complete particles for the next time step.

Figure 2: Algorithm for one time step of partition sampling on the articulated object in Figure 1(a).

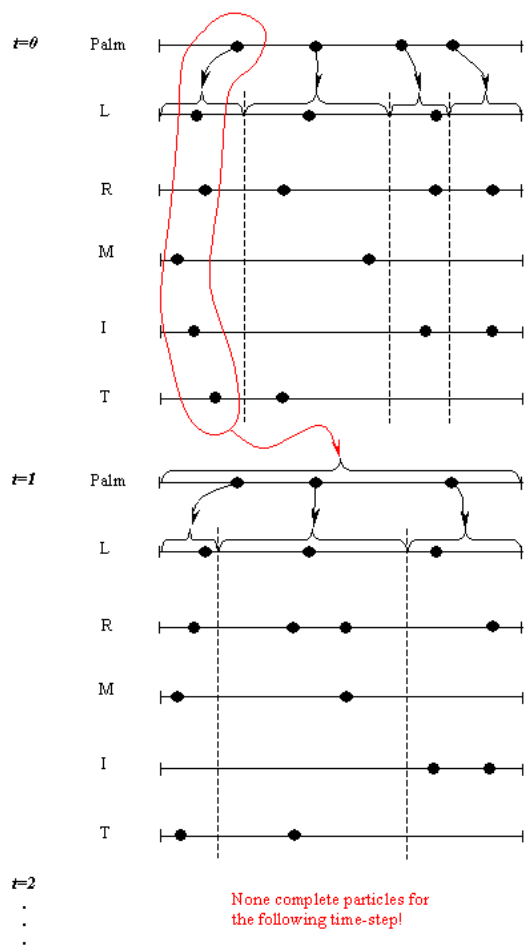


Figure 3: Particle set diagram showing two fictitious time steps of partition sampling for the example articulated hand. At the end of the time step $t=0$ there is one complete particle, circled in red. At the end of the time step $t=1$ there are none complete particles.

A potential problem of the algorithm described in Section 2 is in forming complete particles. The procedure to form a complete particle is to group a

selected particle from the parent particle set, and a selected particle from each of the child particle sets. The selected parent particle, by definition, has a large weight, which means its graphical representation, the hand palm, is better aligned with image features and, consequently, it is a good starting point to search for the fingers. The number of child particles generated from a high weight parent particle will be large, therefore, increasing the chances of finding the fingers. However, in practice, when using this single rule, few complete particles can be formed and propagated, and eventually none complete particles can be formed, resulting in premature termination of tracking.

This situation is illustrated in Figure 3 - the horizontal lines represent particle sets and the selected particles are indicated with black dots on the particle sets. The portions of the finger particle sets that are associated with the same parent particle are referred to as *subsets*, which are separated by vertical dashed lines. This particle set diagram only represents the relationships between particle sets. In practice, each particle set can have different sizes, and the number of selected particles in each set can be much larger. In this diagram a complete particle is defined as: the combination of a selected particle in the palm particle set, and from its associated subset, a selected particle in each of the finger particle sets.

In Figure 3, we can see that at the end of time step $t=0$ there is only one complete particle, encircled in red, which propagates to $t=1$; however, at the end of the time step $t=1$ it is not possible to form any complete particles. This situation worsens if the child particle sets have child particle sets of their own. One way of avoiding incomplete particles is to force at least one selected particle in each subset for each child particle set, for example, the particle with highest weight in the subset. However, this could lead to the selection of particles, in the child particle set, with very low weight, i.e. particles that do not represent properly the relevant link. Another possible solution is to use particle interpolation described in the next section.

3 Particle Interpolation

The idea of particle interpolation involves creating new particles in a child particle set based on existing particles in the same particle set, and those in the parent's particle set. In the particle set diagram of Figure 3, the aim of particle interpolation is to generate a new particle for each subset, and the new particle must have the highest possible weight. For example, the particle with highest weight in a finger particle set represents the finger contour that matches the image features better than others. This particle provides information about where the finger is in the image. Particle interpolation will generate a particle, for each subset, that shares some of the image features of the particle with highest weight in the set. The process is illustrated in the particle set diagram of Figure 4. In this diagram the palm particle set has four selected particles, and therefore there are four subsets in the finger particle sets. The large black dots in each of the finger particle sets represent the particle with highest weight in that set. The smaller red dots in the finger particle sets represent the interpolated particles, one for each subset. The interpolated particles in each particle set are calculated by combining data from the particle with highest weight in the particle set and the parent of each of the subsets, represented by red arrows originating from the particle with highest

weight and ending at the interpolated particle in each subset.

We can see that at the end of the time step for each selected particle in the palm particle set, there will be a complete particle. However, although the interpolated particles have a large weight, the exact weight is not known. In order to form complete particles with a known weight, the interpolated particles need to be weighted. The algorithm in Figure 2 can be updated in order to include partition sampling by substituting the Step 2.4 with the following 3 steps:

2.4. Select the particle with highest weight in the finger particle set.

2.5. Generate a new interpolated particle for each subset, based on the particle selected in step 2.4.

2.6. Weight the interpolated particles.

A simplified version of the articulated hand model of Figure 1(a) can be defined to only include the palm and the little finger. Suppose two particles *A*, and *B* use this hand model, see Figure 5. Particle *A* is formed by two partitions: *Ap* corresponding to the palm, and *Af* corresponding to the finger. Similarly, particle *B* is formed by *Bp*, and *Bf*. Assuming that *Af* is the particle with the highest weight in the finger particle set, and *Bf* is a particle with low weight in the same particle set. The interpolation procedure finds new parameters (length, and angle) for *Bf* in order that it shares some image features with *Af*. The goal

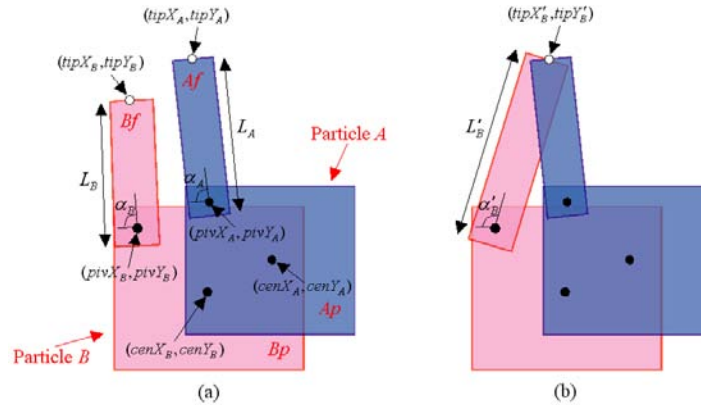


Figure 5: Graphical representation of the interpolation process. (a) Particle *A*'s finger, *Af*, has the highest weight. (b) *Bf* parameters (α_B, L_B) are updated in order that *Bf*'s fingertip coordinates match those of *Af* fingertip.

In Figure 5(a) we can see a graphical representation of particle *A*'s state, and particle *B*'s state. $(cenX_A, cenY_A)$ and $(cenX_B, cenY_B)$ are the palm pivots of *Ap* and *Bp* respectively. $(pivX_A, pivY_A)$ and $(pivX_B, pivY_B)$ are the finger pivots of *Af* and *Bf* respectively. Finger pivots can be calculated from the palm pivots and the palm's state i.e. translation, rotation, and scale. $(tipX_A, tipY_A)$ and $(tipX_B, tipY_B)$ are the fingertips of *Af* and *Bf* respectively. Fingertips can be calculated from the finger pivots and the finger's

of this operation is to maximize *Bf*'s weight, while taking into account the fact that the two particles come from different parents: *Ap*, and *Bp*. A possible rule to maximize *Bf*'s weight in this manner is: *Bf* maximizes its weight if its fingertip coordinates are the same as *Af*'s fingertip coordinates. Other rules are also possible; however, this rule produced the best results in the experiments and is adopted.

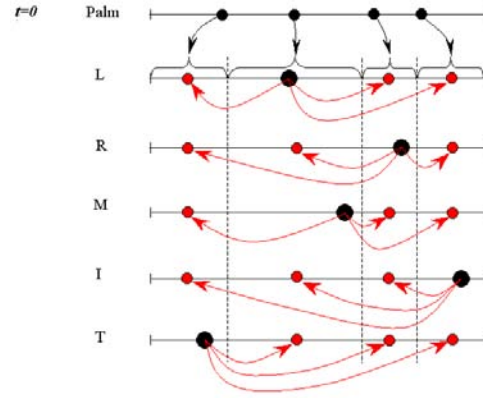


Figure 4: Particle set diagram showing the particle interpolation process. The big black dots in the finger particle sets are the particles with the highest weight in the set. The smaller red dots are the interpolated particles, one for each subset.

state i.e. angle, and length. In order to maximize *Bf*'s weight, its fingertip coordinates must be the same as *Af*'s fingertip coordinates, Figure 5(b). This can be achieved by updating *Bf*'s parameters (α_B, L_B) in the following manner:

$$\begin{aligned} dx &= tipX_A - pivX_B \\ dy &= tipY_A - pivY_B \\ AngleF &= \tan^{-1}\left(\frac{dx}{dy}\right) \end{aligned} \quad (1)$$

$$LengthF = \sqrt{dx^2 + dy^2} \quad (2)$$

$$\alpha'_B = AngleF - OriginalFingerAngle - angle \quad (3)$$

$$L'_B = \frac{LengthF}{(OriginalFingerLength * scale)} \quad (4)$$

Equations (1) and (2) calculate the angle and Euclidean distance between Bf 's pivot and Af 's fingertip. Equations (3) and (4) apply a normalisation to $AngleF$ and $LengthF$ in order that α'_B is relative to the angle of Bp ; and L'_B is a number between 0 and 1. $OriginalFingerAngle$ and $OriginalFingerLength$ are the angle and length of the finger in the template position, i.e. for $\alpha = 0$ and $L = 1$; and $angle$ and $scale$ are the rotation and scale parameters of Bp . Using this rule we can generate new finger particles for any palm particles, and the weight of these new particles is likely to be high.

4 Implementation

This section presents an implementation of an articulated hand contour tracker. The tracker uses partition sampling, as described in Section 2, and particle interpolation, described in Section 3, to track tree-structured objects. This tracker is capable of tracking in real-time the contour of a hand in a video sequence. The tracker can also handle the rigid movement of the hand, and the independent movement of each finger, according to the hand contour model of Figure 6. The model is defined using BSpline curves as described in [3,4,5]. Figure 6(a) shows the control points of the BSpline curves. Figure 6(b) shows the joints of the hand model, which has 14 DOF. Note that the thumb is modeled using two segments, but their lengths are constant. The measurement model is based on [5], and uses both skin colour and edge information.

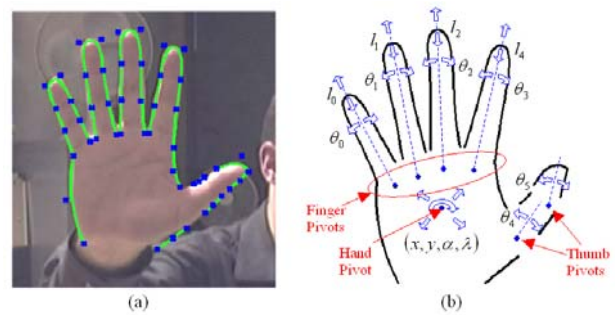


Figure 6: Hand contour model. (a) Hand contour showing 50 control points. (b) Articulated hand contour showing joint parameters.

The tracker is tested with two video sequences. In the first video sequence, a hand is tracked through rigid motions. In the second video sequence, the hand is tracked through a combination of rigid and articulated motion. The performance of the tracker is assessed at each video-frame by measuring the distance between the tracked contour and a ground truth contour, which was calculated manually. This contour distance is calculated using the distance metric defined in [4,6]. The results are shown in Figure 7. The tracker can track successfully both video sequences, with a few occasional incorrect locks of the fingers. The tracker uses only 250 particles in the hand palm particle set, and 100 particles in each of the finger and thumb particle sets. The contour distances for the first video sequence, Figure 7(a), and second video sequence, Figure 7(c), are generally small, with the exception of a few peaks, which are due to the tracker having a temporary lock on the wrong fingers. Four example frames for each video sequence are shown in Figure 7(b), for the first video sequence, and Figure 7(c), for the second video sequence.

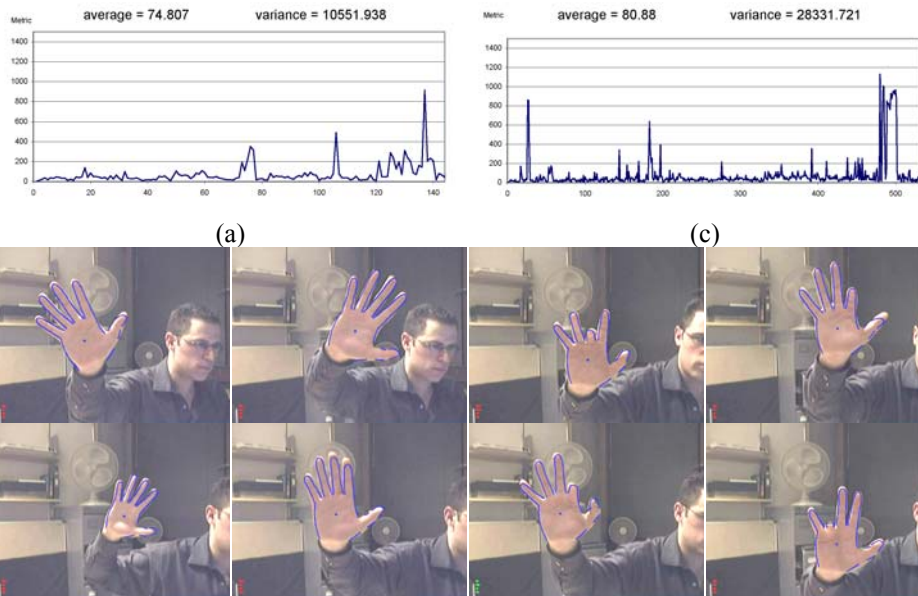


Figure 7: (a) and (c) show hand contour difference against video frames, (b) and (d) show example hand contour tracking from video frames.

5 Conclusion

We have shown how partition sampling can be used with particle interpolation, in order to track tree-structured articulated objects such as hand contours. Particle interpolation is essential to maintain continuity of tracking when a complete particle set cannot be formed. In addition, particle interpolation provides an efficient solution by propagating particles from one time step to the next. The tracker described in this paper improves the method described in [3], which does not take into account the tree structure of the target. The four partitions in which the hand's configuration is divided are dealt with in sequence. With this approach, each time an extra partition is involved in the tracking, the number of particles required increases faster than our approach. Their hand model has 7 DOFs and uses a total of 900 particles - 700 particles for the hand, 100 for the first thumb segment, 10 for the second thumb segment, and 90 for the index finger. Our hand model has 14 DOFs and uses a total of 850 particles - 250 particles for the palm of the hand, 100 particles for each of the fingers segments.

6 References

- [1] Isard, M. and Blake, A. (1998). Condensation - conditional density propagation for visual tracking. *Int. J. Computer Vision*, 28, 1, 5-28.
- [2] MacCormick, J. and Blake, A. (1999). A probabilistic exclusion principle for tracking multiple objects. In *Proc. 7th International Conf. Computer Vision*, 572-578.
- [3] MacCormick, J. and Isard, M. (2000). Partitioned sampling, articulated objects, and interface-quality

hand tracking. In *European Conf. Computer Vision*.

- [4] Blake, A. and Isard, M. (1998). *Active contours*. Springer.
- [5] Isard, M. and MacCormick, J. (2000). Hand tracking for vision-based drawing. Technical report, Visual Dynamics Group, Dept. Eng. Science, University of Oxford. Available from www.robots.ox.ac.uk/~vdg.
- [6] Tissainayagam, P. and Suter, D. (2002). Performance measures for assessing contour trackers. *Int. Journal of Image and Graphics*, 2, 343-359.

Detection and Removal of Global and Local Noise in Realtime Video Streams

A. Clark, R. Green

Department of Computer Science, University of Canterbury, Christchurch.

adrian.clark@hitlabnz.org

richard.green@hitlabnz.org

Abstract

Despite the steady advancement of digital camera technology, noise is an ever present problem with image processing. Low light levels, fast camera motion, and even sources of electromagnetic fields such as electric motors can degrade image quality and increase noise levels. Many approaches to remove this noise from images concentrate on a single image, although more data relevant to noise removal can be obtained from video streams. This paper discusses the advantages of using multiple images over an individual image when removing both local noise, such as salt and pepper noise, and global noise, such as motion blur.

Keywords: image noise, motion blur, salt and pepper, video streams

1 Introduction

Noise is a constant frustration when dealing with computer vision systems. While steps can be taken to minimise the noise, such as using expensive high quality cameras and constraining operating conditions, some noise will still be present. Low quality cameras in unconstrained environments are more commonly being used, and indeed are a more desirable set up for a lot of commercial applications, and these present significant implications for computer vision processing. Despite previous research done in removing noise from video streams[1], the trend is still to treat noise removal on a per image basis[2][3][4].

In this paper, noise is defined to mean artefacts within an image which are the results of inaccuracies in capturing and converting optical information into a digital representation. These artefacts can occur locally, such as a pixel affected by salt and pepper noise, or globally, such as motion blur across an entire image. These two types of noise can be unified as an inverse function of the global ambience. As the global ambience decreases, compounding inaccuracies cause the Signal to Noise Ratio (SNR) to increase, which results in a greater degree of local noise. Subsequently a camera faced with a decrease in global ambience typically will automatically increase the exposure time so that more light can be let in, but an increase in exposure time also elevates the amount of motion blur. Thus a change in a single input factor results in increasing two separate classes of noise.

Figure 1 shows an example relationship tree, showing global ambience, global noise (Motion Blur),

and local noise (Pixel Noise), as well as some common registration techniques, and how they are affected by noise. While both types of noise can be unified under the function, the detection and removal of both is considerably different. For this reason, each type will be discussed separately in the following sections.

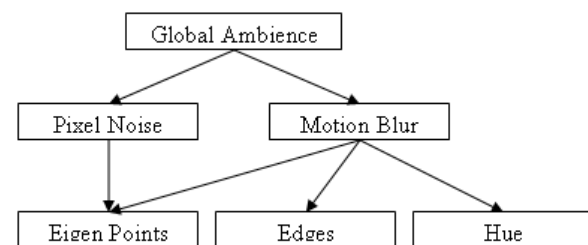


Figure 1: An example of the type of relationships between lighting, noise and registration methods

2 Local Noise

We define local noise as image corruption specific to a certain subsection of an image which is independent of other regions of an image. This leads to a certain amount of “randomness” with the noise, such that the noise content of a pixel cannot be accurately predicted by examining other pixels. The most common types of local noise are Gaussian[5] or salt-and-pepper noise[6]. Salt-and-pepper noise shows up in an image as single pixels with a noticeable difference in colour or intensity from their neighbouring pixels, when in reality there is no discernable difference between the two. Gaussian noise is generally due to a low Signal to Noise Ratio, and as the signal is lower in darker regions of

the image, noise tends to be more prevalent there, as shown in figure 3.

2.1 Calibration

One major advantage of using video as opposed to a single image for noise detection and removal is calibration that can be performed in additional frames. One such approach is to use a banded light diagram like that shown in Figure 2. Such an artificial diagram is computationally simple to find in a video frame, and once found, the variance of illumination can be determined for each light bar. This variance is representative of the noise level for each of the intensities. This can then be used as a method for estimating the likelihood that any given point in future images is noise by examining the intensity of it's neighbouring pixels. A typical example would be a single noisy pixel showing up as a white pixel amongst an area of black pixels. This scenario would provide an excellent interest point to a number of registration algorithms, however by examining the context of the point the system can determine there is a high probability that it is only noise and chose to ignore the point.



Figure 2: The calibration image for calculating noise level at each intensity range.

2.2 Difference of two images

One exploitable characteristic of gaussian noise is that it is randomly distributed. The difference of two consecutive frames will highlight points which have changed between frames, including noise. Any moving objects in the scene will also show up on the difference image, often with a far greater magnitude than noise. To take this relevant difference into account, large difference regions or even high magnitude difference areas can be thresholded. The remaining difference image will show many low intensity pixels which are likely to be caused by noise.

Figure 3 shows a subsection of a negative image (for increased contrast) of an amplified difference image between two frames while both the camera and scene were stationary. The areas where there is greater noise concentration are there regions where the average pixel intensity is lower. Pixels with lower intensity values are more susceptible to noise, just as entire scenes with a lower global ambience also suffer from higher noise levels.



Figure 3: A subsection of a negative image of the difference between two frames. All non-white points are noise

2.3 Detecting Signal to Noise Ratio

Many digital web cameras have automatic white balancing and brightness controls programmed into the firmware, which automatically adjusts the brightness, contrast and exposure time according to light level detected. While this auto adjustment increases the visibility to a human viewing the results, the noise levels are typically increased as well and this can complicate matters when computer vision is used. While it is beneficial to have a consistent brightness level, the method by which this is achieved in the camera results in changing the Signal to Noise Ratio. When the ambient light is reduced, the levels are amplified, and so is the noise.

Unfortunately, many inexpensive digital cameras provide no software facility for retrieving how much light levels have been adjusted, and it is difficult to estimate simply based on changes in the scene. However, by analysing the difference of two images, the frames where the camera has adjusted the brightness level can be determined. An experiment was conducted using a standard USB 2.0 webcam, and it was surprising to notice a stepping effect occurring, where instead of a smooth transition from a low light to normal lighting level, there was a series of sudden increases in lighting level as the automatic gain control compensated for varying ambient light, as shown in Figure 4

The reason for this stepping is unknown, but is assumed that hysteresis is employed to prevent flickering which may occur if the camera was updating the brightness every frame. While the stepping does not give the exact ratio of the actual global brightness compared to the perceived brightness, it does provide the facility to make an assumption about how the ratio may have changed between frames.

2.4 Removing Local Noise

There are a range of methods available for removing noise from an image. Typically noise is removed with a blur or erode[7] filter, to

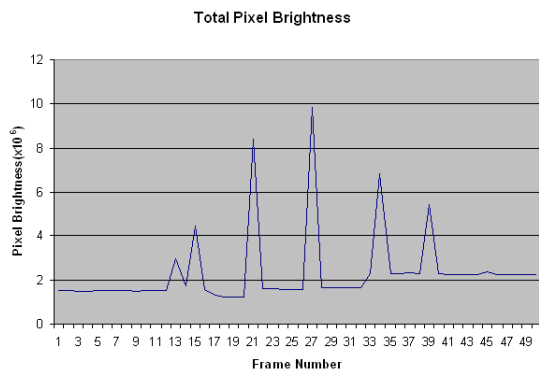


Figure 4: The stepping effect caused by the camera's auto brightness control

average noisy pixels out with neighbouring pixels. However, a global Gaussian filter can remove points which were very important for registration or tracking, as well as reducing the intensity of other significant details for computer vision, such as edges.

One method of resolving this loss of detail involves isolating noisy areas using a filter (typically a median filter), and only blurring a window around that point[8]. The approaches mentioned above can be used as a preprocessing step, as they provide points which are likely to be noisy within images. The thresholded and amplified difference of two frames can be used with a dilate filter to enlarge the area surrounding each of the suspected noisy pixels, and this can then be used as such a filter.

3 Blur

Blur is a problem encountered image processing which can be considered in the same domain as local noise. It is a corruption of image data, and degrades computer vision performance. There are two main types of blur encountered in image processing. One is static blur which can be caused by an out of focus camera, or a damaged camera lens. The other main type of blur is motion blur. Motion blur is often present with motion under low light levels, a problem made worse by the minimal light capture by tiny lenses in cheaper digital cameras.

3.1 Blur Detection

The nature of static blurs such as a damaged camera lens means they do not change properties over successive frames, there is no real advantage of using video over a single image, other than to confirm that it is indeed static. However, motion blur has been examined in video streams in a number

of different ways, such as using two cameras with different optical and temporal resolutions[9],

A variety of algorithms have been designed to remove motion blur, from Wiener filtering to Blind Deconvolution. One common feature of all these blur removal algorithms is that they require some sort of initial estimate of motion blur direction and magnitude to begin the process. While this estimate is not required to be completely, a better initial estimate will yield more accurate results and a faster convergence time. Image analysis can provide an estimate of blur characteristics, but these can often be found far easier and with higher accuracy by analysing the differences between successive video frames using techniques such as Optical Flow to estimate camera direction as shown in figure 5.

In addition to this global approach, individual objects can be segmented based on their spatial relationships[10] or on their direction and magnitude of motion. By doing this, several motion blur deconvolutions can be applied to deblur individual objects within a scene, which results in better performance of the algorithms.

3.2 Blur Removal

An experiment was conducted to compare deblurring when given only a single image, and when given an image sequence. The image sequence is taken by panning (using rotation) a camera in an indoor laboratory environment. The focal subject was a soft toy camel placed a metre in front of the camera. The camera was a standard 640x480 resolution at 25fps USB 2.0 webcam, with the focus set at one metre, so that the subject was in focus. The camera was rotated at approximately 150° per second, so that in the consequent 15 sequential frames, the camera had rotated a full 90°. This provided a reasonable motion blur of the scene.

For the deblurring, scripts were written in Matlab. The single frame deblur required Blind Deconvolution algorithms, while the image sequence used the Wiener filter. A program was written in C++ using the OpenCV library to extract optical flow vectors from the image sequence. These optical flow vectors were grouped according to their direction, as shown in figure 5, the group with the highest count of vectors in it was chosen as the most likely representative of the global direction of motion. These vectors were then averaged to create a single vector to represent direction and magnitude of the motion.

Unfortunately, optical flow velocity does not completely describe motion blur, only an indication of the scale and direction. The true blur depends



Figure 5: Optical Flow vectors, colours show groups based on direction

on exposure time of the camera, which in turn depends on global ambience and camera intrinsics. As a substitute, an initial measurement of blur length was taken and compared to the distance given by optical flow calculations, and this was used as the ratio for calculating blur length purely from optical flow.

3.3 Blur Removal Results

Due to this nature of using a "real" blur as opposed to a computationally added one, there was no "ideal" image for a comparison of the resulting unblurred images. To compare the algorithms, their outputs were compared visually against one another, and an optical flow calculation was performed across two consecutive frames deblurred by each algorithm. The resulting array of vectors from the calculation was filtered based on the angle and magnitude, such that only correct vectors remained. The percentage of vectors which were correct provides a quantitative measure of performance of these algorithms.

It was estimated that the Wiener filter would produce a similar deblurring result in a shorter amount of time than blind deconvolution, due to the extra information available for the deblurring.

3.3.1 Image Sequence Deblurring

The wiener filter was run on the image to be deblurred, using the blurring vector calculated by optical flow to generate the relevant point spread function (PSF). Figure 6 shows the resulting PSF and deblurred image. It appears that there has been a considerable increase in the higher frequency components, especially in low frequency regions, such as the camels chest. While the blurred image showed little detail here, there is

now a considerable amount of finer detail, such that the fur can now be seen.

Unfortunately as a result of the deblurring, parts of the image have begun to "ripple", with high frequency edges spreading out across the image. This is a known effect with Wiener Filters[11], and could be resolved by isolating the areas of deblurring to only lower frequency regions. This has been shown to be successful in previous research, and could prove useful for future applications of this work.

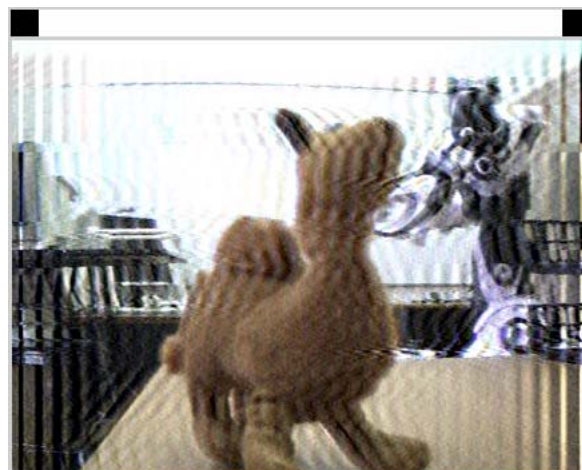


Figure 6: The image after deblurring by the Wiener Filter. The white bar above shows the point spread function calculated.

The time taken for this algorithm to run was less than ten seconds. While this is still not real time, this was unoptimised code running in Matlab's scripting environment, and could likely run faster. The Optical Flow algorithm across two subsequent frames found 233 vectors which matched the angle and magnitude of motion blur found in the image sequence.

3.3.2 Single Frame Deblurring

The blind deconvolution algorithm used was based on the Richardson-Lucy algorithm. As is usually the case, a decrease in information provided to an algorithm will result in a decrease in performance. Blind Deconvolution is designed to be run where there is minimal or no data about the blur. Using this logic, it was expected that Blind Deconvolution would only perform *at best* as good as the Wiener filtering.

The accuracy of Blind Deconvolution depends partly on the estimated size of a calculated Point Spread function. Too large and the point spread function could be overestimated, and the image deblurred too much or even in the wrong direction, too small and the effects of deblurring will be

minimal. To investigate this effect, when the experiment was run for blind deconvolution, two tests were run, to examine the best case, where the estimated size of the PSF function is exactly correct, and the worst case where the estimated size of the PSF function is considerably incorrect. The results are shown in image 7



Figure 7: Left: Best Case Scenario and PSF, Right: Worst Case Scenario and PSF

3.3.3 Single Frame - Best Case

For the best case scenario, the same sized PSF function that was derived from the Wiener Filter was used. For both blind deconvolution trials, two iterations were performed, the first to get an initial estimate of motion, then an intermediary step to attempt to isolate and remove noise in the estimated PSF, finally followed by the second iteration, using the cleaned PSF as input.

Comparing the results of the best case Blind Deconvolution, it appears more or less on par with the results obtained from Wiener Filtering. There appears to be a small increase in detail, but in addition, noise - such as that appearing around the camels eye - has been increased considerably.

The time taken for the best case scenario of Blind Deconvolution was in excess of 450 seconds, far from being realtime. Even with serious optimisation of the code and environment, it is unlikely that this will be a feasible method of removing blur from a live video stream. The optical flow calculation found 248 matching vectors across two deblurred subsequent frames.

3.3.4 Single Frame - Worst Case

The experiment for Worst Case was run using the same code as the best case, apart from the initial estimate of PSF size was a square, the size of the magnitude of the blur. The initial deblurring step here resulted in a PSF where nearly every element had some value in it, but after 50 iterations the PSF was beginning to resemble the correct shape.

As is shown in the point spread function, there appears to be some trend in the direction of blur, however somehow the top corner of the PSF has

Algorithm	Vectors	% Matching	Time(s)
None	476	0.63	0
Wiener	491	0.47	6
Blind - Best	488	0.50	450
Blind - Worst	461	0.45	450

Table 1: Comparison of Algorithms based on Total number of points found, Percentage of found points matching direction of motion, and time taken to perform

become more important. As a result of this large spread, the filter has acted as more like a sharpen filter as opposed to a deblur filter, and the image has the typical artefacts of an image which has been sharpened too much.

The time taken for the worst case scenario of Blind Deconvolution was around the same of the best case, over 450 seconds. The Optical Flow algorithm only found 208 matching points in the worse case deblurring.

3.4 Blur Removal Discussion

The results of the deblurring experiments were not surprising. The wiener filter seemed to provide the best compromise between removing the blur and adding too much noise. While the best case of blind deconvolution had a higher number of matching points, the Wiener filter found a higher number of points in general. The worst case of blind deconvolution, where the PSF was estimated incorrectly, effectively resulted in over-sharpening the image, amplifying noise considerably.

Table 1 shows the results of the three algorithms compared to the unblurred image. Both the Wiener filter and the best case of blind deconvolution resulted in the optical flow algorithm locating more vectors. However the percentage of these vectors matching the motion direction of the camera was lower in both of these than in the original unblurred image, suggesting that perhaps some of these points could be attributed to noise caused by the 'ripple' effect. The worst case deblurring performed worse in both the number of vectors found, and the percentage of these which match the motion of the camera.

Despite the deblurring results for both video streams and single images providing being similar quality wise, the effective time taken to run the filter for video streams was only six seconds, while the added computation of calculating a PSF for the single images required 450 seconds in total. While these algorithms could be tuned to perform better, it could mean the difference between five

and twenty five frames per second, an important difference when real time video is paramount.

4 Conclusion

This paper discusses the detection and removal of noise in video streams. Most previous research has focused on detection and removal only in a single frame, but in doing this useful information has been lost about both the camera and the scene. The results from the experiment would suggest there is validity in processing noise based on an entire video segment, rather than just on a frame by frame basis.

In particular motion blur was looked at in detail, and an experiment carried out to examine the effectiveness and speed of deblurring in both video and single images. It was found that while single image deblurring can still produce results of a similar quality to that of video despite missing important data, the additional time required is considerable, even as much as five times that of deblurring a frame of video when the direction of motion is known.

5 Future Research

The main problem with processing noisy blurred video is the overhead and time taken to do so. Video editing is always time consuming, but in a real time application, this is not feasible. Adaptively varying the number of previous frames to process would help to improve efficiency.

In addition, it would be worthwhile examining other aspects of the video stream to try and find other information which may be useful in increasing the quality of the images.

References

- [1] J. Brailean, R. Kleihorst, S. Efstratiadis, A. Katsaggelos, and R. Lagendijk, "Noise reduction filters for dynamic image sequences: a review," *Proceedings of the IEEE*, vol. 83, no. 9, pp. 1272–1292, 1995.
- [2] A. N. Hirani and T. Totsuka, "Combining frequency and spatial domain information for fast interactive image noise removal," in *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, (New York, NY, USA), pp. 269–276, ACM Press, 1996.
- [3] E. Simoncelli and E. Adelson, "Noise removal via bayesian wavelet coring," in *Image Processing, 1996. Proceedings., International Conference on*, vol. 1, pp. 379–382 vol.1, 1996.
- [4] A. Charnbolle, R. De Vore, N.-Y. Lee, and B. Lucier, "Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage," *Image Processing, IEEE Transactions on*, vol. 7, no. 3, pp. 319–335, 1998.
- [5] K. Rank, M. Lendl, and R. Unbehauen, "Estimation of image noise variance," *Vision, Image and Signal Processing, IEE Proceedings*, vol. 146, no. 2, pp. 80–84, 1999.
- [6] N. H. C. Yung, A. H. S. Lai, and K. M. Poon, "Modified cpi filter algorithm for removing salt-and-pepper noise in digital images," vol. 2727, pp. 1439–1449, SPIE, 1996.
- [7] K. Chinnasarn, Y. Rangsanseri, and P. Thitimajshima, "Removing salt-and-pepper noise in text/graphics images," in *Circuits and Systems, 1998. IEEE APCCAS 1998. The 1998 IEEE Asia-Pacific Conference on*, pp. 459–462, 1998.
- [8] R. H. Chan, C. H. Ho, and M. Nikolova, "Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization.," *IEEE Transactions on Image Processing*, vol. 14, no. 10, pp. 1479–1485, 2005.
- [9] M. Ben-Ezra and S. K. Nayar, "Motion-based motion deblurring," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 689–698, 2004.
- [10] S. K. Kang, J. H. Min, and J. K. Paik, "Segmentation-based spatially adaptive motion blur removal and its application to surveillance systems," in *Image Processing, 2001. Proceedings. 2001 International Conference on*, vol. 1, pp. 245–248 vol.1, 2001.
- [11] F. Jin, P. Fieguth, L. Winger, and E. Jernigan, "Adaptive wiener filtering of noisy images and image sequences," in *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, vol. 3, pp. III–349–52 vol.2, 2003.

Matching Moving Objects by Parts with a Maximum Likelihood Criterion

Eric Dahai Cheng and Massimo Piccardi

Faculty of Information Technology, University of Technology, Sydney (UTS),
PO Box 123, Broadway NSW 2007, Australia

Email: {cheng, massimo}@it.uts.edu.au

Abstract

In this paper we present an algorithm for matching the appearance of two moving objects based on a matching-by-parts approach and a maximum likelihood criterion. We assume that the two moving objects to be matched are first extracted from videos by a preliminary foreground extraction-tracking step and our goal is that of matching their appearances. To this aim, we first consider the matching between single frames, one from each track. In order to increase the ability of discriminating between two different physical objects while keeping the matching rate of a single physical object high, each object is divided into N parts and then parts are matched in pairs. The appearance of each part is represented by a colour histogram (called MCSHR for short in the following) and a histogram similarity measure is used to compare two parts. The single-frame matching result is then obtained by fusing the similarities of each part matching. Later, our track matching algorithm extends the single-frame matching along the objects' tracks by a post-matching integration algorithm. Experimental results presented in this paper show that the proposed similarity measurement is accurate at the single-frame level and that the post-matching integration makes the overall matching more robust and reliable.

Keywords: Moving object tracking, object matching by parts, maximum likelihood criterion, major colour spectrum histogram representation, colour distance, similarity measurement.

1 Introduction

Robustly tracking a single object throughout a network of cameras is an important function for effective video surveillance of wide areas [1-4]. However, in most real-world camera networks it is not possible to track a moving object through a continuity of overlapping camera views. Instead, most often the object has to completely exit from the view of a certain camera before it can reappear under the view of a different one. This common scenario is often referred to as disjoint camera views, where observations of a single object are disjoint in time and space to a certain extent. In order to allow tracking in such a scenario, single-camera views of a same object must be matched across neighbouring cameras.

In the following, we assume that each object is extracted and tracked within each single camera view by a preliminary foreground extraction-tracking step, and that the relevant information (the object's blob in each frame) is available – hereafter we call such sequence of blobs *track* for simplicity. Our goal is then that of matching tracks from disjoint views by using some objects' appearance features. To this aim, in this paper we present an algorithm for appearance matching based on a matching-by-part approach and a maximum likelihood criterion. First, we choose the two tracks to compare and consider the first frame in

each. We compare the blobs from these two frames by dividing each blob into N parts, and orderly comparing parts in pairs. Each pair matching provides a similarity measurement, or matching belief, bounded between 0 and 1. The N results from part matching are then fused by an average rule and compared against a threshold set based on a maximum likelihood criterion to provide the results at the frame level. The single-frame matching is repeated for following frames in the tracks and, eventually, such results are integrated to obtain the overall matching result between the two tracks.

The appearance representation is based on a colour histogram with sparse bins. A colour distance based on a geometric distance between two points in the RGB space is first introduced to measure the similarity of any two colours. By using the colour distance and a given threshold, pixels from each part are clustered into a limited number of bins, with each bin's frequency defined as the number of pixels falling into that bin. Such bins are then sorted in descending frequency order and a chosen percentage of them (in our work, 90%) is chosen as major colours to represent the part's appearance. We call this histogram the major colour spectrum histogram representation (MCSHR). A criterion is then defined to assess the similarity, bounded between 0 and 1, of the MCSHRs of two given parts.

To date, the problem of matching the appearance of objects across disjoint camera views has been addressed in relatively a few papers in the literature; [5] and [6] are notable examples. In both [5] and [6], the matching based on appearance is reinforced by the use of priors based on statistics on travelling times between cameras acquired during a learning stage. The main problem that we identify with such an approach is that matching is more prone to fail in anomalous cases, which are instead those of interest for surveillance. For instance, if people remain in a blind area for long time in order to carry out activities such as tampering or stealing, their re-appearance under camera views will occur outside of statistical timing windows. For this reason, our approach deliberately avoids the use of time-based priors. Moreover, unlike [1], [5], [6], we use a part-based matching that prevents false matches between people with similar overall colours but with different spatial distribution.

2 Maximum Likelihood Criteria for Moving Objects Matching by Parts

2.1 Feature Space and Maximum Likelihood Criteria

The raw feature vectors in the observation space of the two matching moving objects are the major colours of the divided parts, shown in the following equations:

$$X_1 = [X_{11}, X_{12}, \dots, X_{1N}] \quad (1)$$

$$X_2 = [X_{21}, X_{22}, \dots, X_{2N}] \quad (2)$$

where X_{1i} and X_{2i} are major colour vectors of the i th divided parts in moving objects one and two, and N is the number of divided parts. Since the major colour vectors are multiple dimensional vectors, their distributions are very difficult to estimate. Therefore, the similarity between two matching objects is used as an observation variable (or one dimensional space) in the process of deriving an optimum matching structure based on maximum likelihood criteria.

The hypothesis test assumes:

$$\begin{aligned} H_0 : sim &= \mu_0 + n, \\ H_1 : sim &= \mu_1 + n. \end{aligned} \quad (3)$$

where n is the error noise that produced in the process of major colour similarity calculation, and based on experience of our experiments, we believe that the noise has Gaussian probability distribution with zero mean and variance σ_N^2 , i.e. $n \in N(0, \sigma_N^2)$; μ_0 and μ_1 ($\mu_1 > \mu_0$) are the average similarities when H_0 (objects are two physically different objects) and H_1 (objects are a single physical object) are true. For simplicity, we assume that μ_0 and μ_1 are constant and

that variations are to be blamed on the noise component.

The above assumptions can be validated by testing the data reported in Tables 1 (for μ_0) and 2 (for μ_1) in Sections 5.1 and 5.2, respectively. In this case, $\mu_0 = 0.4638$ and $\mu_1 = 0.7843$ and assumption $\mu_1 > \mu_0$ is verified. The standard deviations are $\sigma_0 = 0.046$ and $\sigma_1 = 0.056$. In the following, since their difference is small, we treat them as a same value.

Thus, the probability distribution function of sim under the hypothesis of H_0 , is shown in equation (4).

$$p_0(sim) = \frac{1}{\sqrt{2\pi}\sigma_N} \exp\left[-\frac{(sim(X_1, X_2) - \mu_0)^2}{2\sigma_N^2}\right] \quad (4)$$

Similarly, the probability distribution function of sim , under the hypothesis of H_1 , is shown in equation (5).

$$p_1(sim) = \frac{1}{\sqrt{2\pi}\sigma_N} \exp\left[-\frac{(sim(X_1, X_2) - \mu_1)^2}{2\sigma_N^2}\right] \quad (5)$$

The likelihood ratio (LRT) is calculated as follows:

$$\Lambda(sim(X_1, X_2)) = \frac{P(sim(X_1, X_2) | H_1)}{P(sim(X_1, X_2) | H_0)} > \eta \quad (6)$$

Taking the natural logarithm of both sides of equation (6) to obtain the log LRT:

$$\begin{aligned} \ln \Lambda(sim(X_1, X_2)) & > \ln \eta \\ & = \frac{(2sim(X_1, X_2) - \mu_0 - \mu_1)(\mu_1 - \mu_0)}{2\sigma_N^2} > \ln \eta \end{aligned} \quad (7)$$

Equation (7) can be simplified as:

$$sim(X_1, X_2) > \left(\frac{\mu_1 + \mu_0}{2} + \frac{2\sigma_N^2 \ln \eta}{\mu_1 - \mu_0} \right) = \lambda \quad (8)$$

The above equation shows the optimum structure of the matching detector, in which the optimum threshold is the function of the average similarities - μ_0 , μ_1 , and the variations of similarity - σ_N^2 . In the sense of maximum likelihood criteria, in order to minimize the total error (detection and false alarm) η should be 1, so the optimum threshold in equation

$$(8) \text{ becomes } \lambda = \left(\frac{\mu_1 + \mu_0}{2} \right).$$

2.2 Matching Performance Evaluation

Just as a corollary, we show in the following that the matching performance can be easily evaluated in terms of the probability of detection - P_D and false

alarm rate - P_{fa} - as a function of the average similarities, μ_0 , μ_1 , and variance σ_N^2 .

The probability density functions (pdf) of matching objects under H_0 and H_1 described in equations (4) and (5) are shown in Fig. 1.

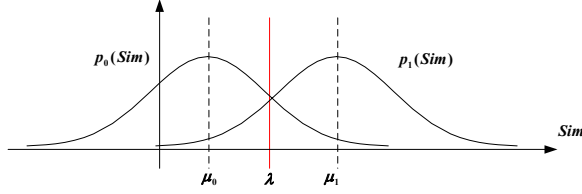


Figure 1: Similarity probability density functions.

In Fig. 1, the probability of false alarm matching - P_{fa} is the area under function $p_0(sim)$ above the detection threshold - λ , i.e.

$$P_{fa} = \int_{\lambda}^{\infty} p_0(sim) dsim \quad (9)$$

$$= \int_{\lambda}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_N} \exp\left[-\frac{(sim(X_1, X_2) - \mu_0)^2}{2\sigma_N^2}\right] dsim = \alpha$$

The probability of the detection or correct matching - P_D - is the area under function $p_1(sim)$ above the detection threshold - λ , i.e.

$$P_D = \int_{\lambda}^{\infty} p_1(sim) dsim \quad (10)$$

$$= \int_{\lambda}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_N} \exp\left[-\frac{(sim(X_1, X_2) - \mu_1)^2}{2\sigma_N^2}\right] dsim = \beta$$

Thus, equations (9) and (10) show that the probabilities of correct matching and mismatching are simple functions of the average similarities, μ_0 , μ_1 , and variance σ_N^2 .

3 Major Colour Spectrum Histogram

3.1 Concept of Colour Distance

In this section, we first introduce the concept of ‘‘colour distance’’ between two colour pixels in the RGB space based on a normalized geometric distance between the two pixels. Such a geometric distance is defined in equation (11):

$$d(C_1, C_2) = \frac{\|C_1 - C_2\|}{\|C_1\| + \|C_2\|} = \frac{\sqrt{(r_1 - r_2)^2 + (g_1 - g_2)^2 + (b_1 - b_2)^2}}{\sqrt{r_1^2 + g_1^2 + b_1^2} + \sqrt{r_2^2 + g_2^2 + b_2^2}} \quad (11)$$

C_1 and C_2 are the colour vectors. The smaller the colour distance, the more similar are the two colours.

3.2 Moving Object Major Colour Representation

By using the concept of colour distance, we can scale down the possible colours to a very limited number of ‘‘major colours’’ (for example, several hundreds) without losing much accuracy on representing a

moving object. For each part of a moving object, a given certain percentage of major colours are retained in the representation, while colours that rarely appear are discarded [7, 8]. Colours within a given mutual distance threshold are dealt with as a single colour. An example picture (‘‘tn_flower’’) is shown in Fig. 2 (a) in which we can see that the most frequent colours are around dark green-black and yellow values. Fig. 2 (b) shows that the histogram of the major colours (under the colour distance threshold of 0.01) seems a faithful representation of the image’s colours and their frequencies.

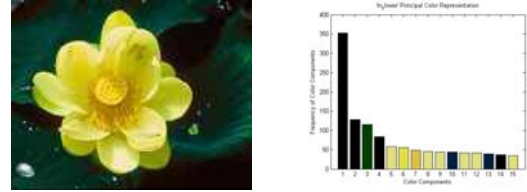


Figure 2. The Major Colour Spectrum Histogram Representation (MCSHR) of the ‘‘tn_flower’’.

4 Single-Frame Matching and Post-Matching Integration Algorithm

4.1 Moving Objects Parts Similarity Measurements

In this section, a similarity measurement based on a most-similar-colour searching algorithm is proposed to measure the similarity between two corresponding parts of moving objects. This algorithm is based on the major colour spectrum histogram of the two corresponding divided parts of the two moving objects. We assume that there are M major colours in the spectrum of a divided part in a moving object A , which can be represented as:

$$MCSHR(A) = \{C_{A_1}, C_{A_2}, \dots, C_{A_i}, \dots, C_{A_M}\} \quad (11)$$

where $C_{A_i}, i = 1, 2, \dots, M$ is the colour vector (RGB) of the major colours in the divided part in object A . The major colour frequencies of the divided part in object A ’s can be represented as:

$$p(A) = \{p(A_1), p(A_2), \dots, p(A_i), \dots, p(A_M)\} \quad (12)$$

The major colour spectrum histogram of object B can be represented similarly. In order to define the similarity of the corresponding divided parts of two moving objects, for each color C_{A_i} in A , a corresponding colour, $C_{B_j|A_i}$, is searched in B as:

$$C_{B_j|A_i} = \arg \min_{k=1, \dots, L} \{d(C_{B_k}, C_{A_i}) < \sigma\} \quad (13)$$

$C_{B_j|A_i}$ is the closest colour to C_{A_i} within a threshold, σ , and $p^{[A_i]}(B_j)$ its frequency. Then, the similarity of C_{A_i} and $C_{B_j|A_i}$ is defined as:

$$Similarity(C_{A_i}, C_{B_j|A_i}) = \min\{p(A_i), p^{[A_i]}(B_j)\} \quad (14)$$

The similarity of the divided part in object A and its corresponding part in object B is obtained by adding up over $i = 1, 2, \dots, M$:

$$\text{Similarity}(A, B) = \sum_{i=1}^M \text{Similarity}(C_{A_i}, C_{B_j|A_i}) \quad (15)$$

In a similar way we can obtain $\text{Similarity}(B, A)$ that generally differs from $\text{Similarity}(A, B)$ since the colour pairs defined by (13) may be different in the two directions. However, if A and B are the same physical object, these two similarities would be approximately symmetric. Therefore, in the final matching criterion we give importance to the symmetry of $\text{Similarity}(A, B)$ and $\text{Similarity}(B, A)$. We first define:

$$\text{Similarity}_{\min} = \min \{ \text{Similarity}(A, B), \text{Similarity}(B, A) \} \quad (16)$$

$$\text{Similarity}_{\max} = \max \{ \text{Similarity}(A, B), \text{Similarity}(B, A) \} \quad (17)$$

Then, we combine them into a single final value, $\text{Similarity}_{A, B}$:

$$\text{Similarity}_{A, B} = \begin{cases} \text{Similarity}_{\min} & \text{if } \text{Similarity}_{\min} < \eta_{\text{discrim}} \\ 1 - \frac{\text{Similarity}_{\max} - \text{Similarity}_{\min}}{\text{Similarity}_{\max} + \text{Similarity}_{\min}} & \text{otherwise} \end{cases} \quad (18)$$

If Similarity_{\min} is lower than a discrimination threshold, η_{discrim} , we bound $\text{Similarity}_{A, B}$ to it. Instead, if Similarity_{\min} is above or equal the discrimination threshold, we choose to check the difference between the maximum and minimum similarities in a ratio form for asymmetry. The bigger the difference between the maximum and minimum similarities, the lower is $\text{Similarity}_{A, B}$. Eventually, matching is assessed if $\text{Similarity}_{A, B}$ is above an assigned similarity threshold.

4.2 Similarity at the Whole-Object Level

Once obtained a similarity value, bounded between 0 and 1, for each pair of divided parts, the values for all the N part pairs need to be combined in order to obtain a single matching result at the whole-object level. For this, one can choose amongst popular fusion techniques such as the product rule, average rule or weighed average rule [9]. The product rule suffers from the famous ‘‘curse of product’’ and should be used only in the case of complete statistical independence between the values to be fused. In our application, some degree of correlation instead certainly exists (two adjacent parts may share parts of a same piece of clothes and thus be materially correlated; the body shape deformats along the sequence, hence blob parts map on different bodily parts along frames) and therefore we cannot use the product. The weighted average rule requires a very well informed estimation of weights to be likely to outperform the (unweighted) average rule [9].

Therefore, we chose to use the latter in our approach. Equation (19) provides the required similarity at the whole-object level.

$$\text{sim}(X_1, X_2) = \frac{1}{N} \sum_{i=1}^N \text{sim}(X_{1i}, X_{2i}) \quad (19)$$

4.3 Single-Frame Matching and Post-Matching Integration Algorithm

In the track matching algorithm, we consider the same number of frames from each track. Moving objects from corresponding frames in Track One and Track Two are matched based on similarity of their major colour spectrum, and the matching results are given as a binary decision.

The second step is the multi-frame post-integration, normalization, and thresholding. The advantages of this algorithm are:

- The single-frame matching is based on the major colour spectrum histogram and two direction similarities measurements, which makes the single-frame matching very accurate.
- The final conclusion is made based on the statistical average of single-frame matching. So, no detailed feature errors are carried forward after this stage, which makes the track matching conclusion more reliable than single frame matching.

5 Experimental Results and Analysis

In our experiments, we report example results from three typical tracks from the PETS 2001 dataset where moving objects have been detected and tracked. The segmented moving objects, major colour spectrum histograms and experimental results are shown in the following sections.

5.1 Matching-by-parts of Two Different Moving Objects

The first case reported here are from two different persons (track 1, frames 0400-0412 and track 2, frames 2150-2162), with two sets of typical extracted moving objects and object masks shown in Figure 3.

In the test, the moving objects are equally divided into seven parts along the vertical direction. The results from matching-by-parts at the single frame level and post-matching integration along the track with 90% of major colours, colour threshold of 0.01, discrimination threshold of 0.35, and matching threshold of 0.6241 are shown in Table 1. While other thresholds are empirical, the matching threshold, λ , is calculated as in equation (8) based on Tables 1 and 2. The results in Table 1 shows that all seven cases are correctly discriminated, with similarities at the whole-

object level between 0.41 and 0.55, and the post-integration rate of 0%. Thus, the two tracks are reliably discriminated.



(1a) MO and mask in frames 0400 and 2150

Figure 3. Moving objects from track 1, frames 0400-0412 and track 2, frames 2150-2162.

Table 1. Matching similarities. (PETS 2001 dataset 1, frames 0400-0412 and 2150-2162, Color distance: 0.01, discrimination threshold: 0.35, MCSHR cut off: 90%, Number of divided parts: 7).

Test Case	Frame No	Part 1	Part 2	Part 3	Part 4	Part 5	Part 6	Part 7	Similarity (mean)	Matching Results
1	0400	0.6441	0.1823	0.1640	0.3388	0.8584	0.9225	0.0455	0.4508	0 (No)
	2150									
2	0402	0.7267	0.8094	0.2492	0.2133	0.2217	0.8907	0.7669	0.5540	0 (No)
	2152									
3	0404	0.2639	0.2999	0.1933	0.2234	0.3176	0.8233	0.7481	0.4099	0 (No)
	2154									
4	0406	0.1724	0.1889	0.1135	0.3267	0.8131	0.8657	0.6451	0.4465	0 (No)
	2156									
5	0408	0.2542	0.7222	0.2024	0.1705	0.3207	0.8100	0.9404	0.4886	0 (No)
	2158									
6	0410	0.2361	0.2152	0.1739	0.2480	0.7907	0.7070	0.7414	0.4446	0 (No)
	2160									
7	0412	0.9398	0.1681	0.0705	0.7958	0.6600	0.2402	0.2933	0.4525	0 (No)
	2162									
Post-Integration	0400-0412									0% (No)
	2150-2162									

5.2 Matching-by-parts of a Single Moving Object in Two Different Tracks

The test data reported here is from the same person in two different tracks (track 1, frames 2040-2052, and track 2, frames 2150-2162 in steps of five frames). The extracted moving object and moving object mask in typical frames (2048 in track 1, and 2156 in track 2) are shown in Fig. 4. The results from matching-by-parts at the single frame level and post-matching integration along the track with 90% of major colours, colour threshold of 0.01, discrimination threshold of 0.35, and matching threshold of 0.6241 are shown in Table 2. The results in Table 2 show us that in all seven cases, similarities were between 0.70 and .87, proving that the proposed matching-by-parts

MCSHR algorithm offers an accurate appearance representation and similarity measurement. The post-integration of the seven individual matching cases is 1.0, thus the two tracks are reliably matched.



(a) MO and mask in frame 2048 (b) MO and mask in frame 2156

Figure 4. Moving objects from track 1, frames 2040-2052 and track 2, frames 2150-2162.

Table 2. Matching similarities. (PETS 2001 dataset 1, frames 0400-0412 and 2150-2162, Color distance: 0.01, Discriminate threshold: 0.35, MCSHR cut off: 90%, Number of divided parts: 7).

Test Case	Frame No	Part 1	Part 2	Part 3	Part 4	Part 5	Part 6	Part 7	Similarity (mean)	Matching Results
1	2040	0.2681	0.6239	0.8946	0.8581	0.7363	0.6118	0.9360	0.7041	1 (Yes)
	2150									
2	2042	0.7146	0.7568	0.9188	0.9344	0.9517	0.3098	0.6402	0.7466	1 (Yes)
	2152									
3	2044	0.7817	0.9093	0.9493	0.8878	0.9280	0.8268	0.8428	0.8751	1 (Yes)
	2154									
4	2046	0.8171	0.6491	0.8222	0.9308	0.8707	0.7943	0.7867	0.8101	1 (Yes)
	2156									
5	2048	0.6718	0.7190	0.8564	0.9283	0.9626	0.7997	0.3050	0.7490	1 (Yes)
	2158									
6	2050	0.8139	0.6633	0.6983	0.9550	0.8936	0.8258	0.7009	0.7930	1 (Yes)
	2160									
7	2052	0.7588	0.5820	0.8044	0.9618	0.9216	0.7687	0.8879	0.8122	1 (Yes)
	2162									
Post-Integration	2040-2052									100% (Yes)
	2150-2162									

6 Conclusions

In this paper, a matching-by-parts algorithm based on maximum likelihood criteria has been proposed. Based on our experimental results, the following conclusions can be drawn:

- 1) The proposed moving object matching-by-parts algorithm shows both good invariance and discrimination.
- 2) The assumptions made in the model in (3) are well validated by results reported in Tables 1 and 2. This allows formal derivation of the matching threshold, λ .
- 3) Thanks to the post-matching integration, potential single-frame matching errors do not affect the overall matching result and robustness and accuracy are increased.

The proposed moving objects track matching-by-parts algorithms can significantly extend current video surveillance applications by providing them with the capability of tracking single objects across disjoint camera views which is the actual case for many real-world surveillance camera networks.

7 Acknowledgements

This research is supported by the Australian Research Council, ARC Discovery Grant Scheme 2004 (DP0452657).

8 References

- [1] M. Piccardi and E. D. Cheng, "Track Matching Over Disjoint Camera Views Based on an Incremental Major Color Spectrum Histogram", IEEE Int. Conf. on Advanced Video and Signal based Surveillance, Como, Italy. (AVSS 2005)
- [2] T.H. Chang and S. Gong, "Tracking Multiple People with a Multi-Camera System", Proceedings of the 2001 IEEE Workshop on Multi-Object Tracking, 19-26, 2001.

- [3] A. Lipton, H. Fujiyoshi, and R. Patil, "Moving target classification and tracking from real-time video," in Proc. Of the IEEE Image Understanding Workshop, 1998, pp. 129-136.
- [4] A. Elgammal, D. Harwood, and L. Davis. "Non-parametric Model for Background Subtraction", 6th European Conference on Computer Vision, 2000.
- [5] O. Kaved, Z. Rasheed, K. Shafique, M. Shah, "Tracking Across Multiple Cameras With Disjoint Views," in Proc. of the Ninth IEEE Int. Conf. on Computer Vision (ICCV'03), vol. 2, pp. 952-957.
- [6] D. Makris, T. Ellis, and J. Black, "Bridging the Gaps between Cameras," in Proc. of the 2004 IEEE CS Conf. on Computer Vision and Pattern Recognition, vol. 2, pp. 205-210.
- [7] W. Lu and Y. P. Tan, "A Color Histogram Based People Tracking System", Proc. IEEE Int'l Symp. Circuits and Systems, vol. 2, pp. 137-140, 2001.
- [8] Liyuan Li, Weimin Huang, I.Y.H. Gu, K. Leman, Qi Tian, "Principal Color Representation for Tracking Persons," in Proc. of IEEE Int. Conf. on Systems, Man and Cybernetics 2003, vol. 1, pp. 1007-1012.
- [9] F. Roli, G. Fumera, "A theoretical and experimental analysis of linear combiners for multiple classifier systems, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 27, Issue 6, Jun 2005, Pages 942 - 956

Semi-supervised Silhouette Detection for Thermal Imaging

§Surya Prakash and §†Antonio Robles-Kelly

§RSISE, Bldg. 115, Australian National University, Canberra ACT 0200, Australia

†National ICT Australia (NICTA), Locked Bag 8001, Canberra ACT 2601, Australia
{surya.prakash, antonio.robles-kelly}@anu.edu.au

Abstract

In this paper, we describe the use of a voting scheme and discriminant analysis for purposes of the recovery of silhouettes from a set of sequentially acquired images. The method presented here is semi-supervised in nature and employs, as a starting point, a manually obtained silhouette. By assuming the camera displacement is small between successive frames, we cast the silhouette recovery problem in a discriminant analysis context, in which the initial silhouette information provided by the user is propagated from frame-to-frame across the image sequence. To this end, we make use of a region-based voting scheme. This, in turn, allows us to pose the silhouette recovery task in an evidence combining setting. We illustrate the utility of the method on real-world thermal imagery.

Keywords: silhouette detection, object contour, discriminant analysis, voting approaches

1 Introduction

Silhouette detection and recovery is a classical problem in computer vision, pattern recognition and computer graphics, which arises in a number of applications spanning from rendering and shadow modelling [1] to articulated body kinematics [2] and volume modelling [3].

A silhouette can be viewed as an image in which the object becomes an occluder of the background from the observer’s view point. In the computer graphics community, the recovery of object silhouettes is a forward computation process which can be effected making use of computations which hinge in the properties of three-dimensional data. These methods are aimed at recovering the boundary of the object based upon ray tracing or image processing operations. Therefore, the silhouette is computed at the image rendering level. Raskar and Cohen [4] have used polygon orientation on the mesh under study to recover the silhouette of the object. Kettner and Welzl [5] used a sweep-line algorithm for the computation of the silhouette of a polyhedral object.

Unfortunately, in the areas of computer vision and pattern recognition, the recovery of the object silhouette is a “backwards” computation task in which the occluding contour is computed making use of image processing techniques rather than computations in 3D. This image space silhouette computation is not a straightforward task. The reason being is that, in contrast with the use of 3D information, here, we are concerned with the

recovery of the projection, on the image plane, of an occluding contour in the scene. Thus, in general, when 3D information is not at hand, the recovery of silhouettes is dependent on background information and controlled lighting conditions.

The link between the occluding contour and the 3D structure of the object under study has motivated the use of silhouettes in the computer vision community as a cue for shape recovery, i.e. shape-from-silhouette. These approaches generally rely upon the manual generation of the object silhouettes [6] or the recovery of the occluding contour via image differencing [7]. This is done making use of a known background so that the foreground and background are distinguishable. Along these lines, Zheng [8] has recovered the 3D shape from a single camera using multiple views of the object under study on a turntable with known background. In a related development, Zeng and Quang [9] use multiple calibrated images to overcome the background modeling problem associated to silhouette recovery.

In this paper, we aim at recovering the silhouettes of the object under study making use of a number of images acquired using an uncalibrated camera and a single contour provided by the user. This is, for each of the images under study, our aim of computation is the corresponding occluding contour with respect to the background, whose nature is, *a priori*, unknown. This silhouette can be viewed as a contour whose recovery is based upon a semi-supervised process governed by the single

silhouette provided at input by the user, which is propagated across the views at hand.

2 Semi-supervised Silhouette Extraction

In this section, we present our silhouette extraction algorithm. The method is, succinctly, as follows. For every pair of views, between which the camera displacement is small, we cast the silhouette recovery problem as a two-class segmentation one, where the contour is given by the boundary pixels between the object under study and the background. To do this, we make use of a discriminant analysis on the background and foreground information whose starting point is the user-provided contour. Hence, our silhouettes recovery algorithm is a semi-supervised one in which the contour pixels are recovered using a separability analysis approach based upon a two-class characterization of the problem in hand.

The section is organized as follows. We commence by viewing the silhouette of the object as a contour on the image plane which is the result of the projection onto the camera frame of a set of 3D points in the scene. Viewed in this manner, we can show that, if the displacement of the camera is small, the pixel coordinates for each of the 3D points in the scene are approximately the same. This, in turn, permits the use of a region-based approach and discriminant analysis [10] to cast the silhouette recovery problem in a clustering setting. We conclude the section by showing how the region information and the discriminant analysis between foreground and background can be combined in a probabilistic fashion so as to recover the silhouette of the object in the scene.

2.1 Silhouette Modeling

As mentioned earlier, an object silhouette is the occluding contour of the object. Here, we aim at recovering the silhouette information from a sequence of views. To do this, we commence by viewing the pixel information as the projection onto the camera frame of a set of points in 3D. Thus, let $P_i = (X_i, Y_i, Z_i)$ be the 3D point on the object referenced to the camera frame indexed i , i.e. the i th view. Making use of the pin-hole camera model [11], we can write the normalized image projections, $\mathcal{P}_n^i = [x, y]$ as follows

$$\mathcal{P}_n^i = \begin{bmatrix} X_i/Z_i \\ Y_i/Z_i \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \quad (1)$$

After including the lens distortion, the new normalized point coordinate $\mathcal{P}_d^i = [x_d^i, y_d^i]$ can be re-

lated to the pixel image coordinates, x_p^i and y_p^i as follows

$$\begin{bmatrix} x_p^i \\ y_p^i \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} x_d^i \\ y_d^i \\ 1 \end{bmatrix} \quad (2)$$

where \mathbf{K} is known as the camera parameter matrix.

Now consider the same 3D point on the camera frame indexed $i + 1$, the i th+1 view. The relationship between P_i and P_{i+1} can be written as

$$P_i = \mathbf{T} + \mathbf{R}P_{i+1} \quad (3)$$

where \mathbf{R} and \mathbf{T} are the relative rotation and translation of $i+1$ with respect to i . Further, as outlined above P_{i+1} can be projected to its image plane using Equation 1.

However, should the relative positions of the camera frames $i + 1$ and i be close to one another, we can rewrite equation 3 as

$$P_i \approx P_{i+1} \quad (4)$$

Consequently, if the camera distortion and position with respect to the object under study are approximately the same between successive views, then we can write

$$\begin{bmatrix} x_p^i \\ y_p^i \end{bmatrix} \approx \begin{bmatrix} x_p^{i+1} \\ y_p^{i+1} \end{bmatrix} \quad (5)$$

In other words, the neighbourhood of (x_p^i, y_p^i) will be similar if not exactly same as that of (x_p^{i+1}, y_p^{i+1}) . In the following section, we show how this can be exploited for purposes of separating the background from the foreground using the user-input silhouette.

2.2 Silhouette Recovery as a Foreground-Background Separation

As mentioned above, if the displacement between views is small, then the information conveyed by the neighbourhoods for the pixel coordinates x_p^i and x_p^{i+1} is approximately equivalent. This implies that the labelling of background and foreground extracted from the user-input silhouette first frame in our sequence can be propagated across the views under study making use of an appropriately sized neighbourhood window. Further, once the silhouette for the i th view is at hand, we can use it, analogously to the user-input one at the first frame, to perform inference on the view indexed $i + 1$. In this section, we first show how the pixels in the frame $i + 1$ can be classified, regionwise, using the silhouette for the i th view. With this region-based information at hand, we then recover the silhouette for the $i + 1$ frame by making use of a

simple discriminant analysis method and a voting scheme.

Consider a region $\mathcal{R}_{\hat{p}}^i$ in the image corresponding to the i th view given by a sliding window centered at the pixel \hat{p} . Making use of the foreground-background separation provided by the silhouette for the i th frame, we can classify the pixels in the region $\mathcal{R}_{\hat{p}}^{i+1}$ on the frame indexed $i+1$. Note that both, $\mathcal{R}_{\hat{p}}^i$ and $\mathcal{R}_{\hat{p}}^{i+1}$ are centered at the same pixel coordinates and, hence, overlap. Moreover, from now on, we consider the coordinates on the image plane for the pixels p^i and p^{i+1} equivalent. Once the pixel classification for each region is at hand, we can use an evidence combining scheme in which, for any given pixel in the image, there will $|\mathcal{R}_{\hat{p}}^i|$ regions voting for the pixel p^{i+1} to be considered either foreground or background. Once the votes are at hand, we can use a method akin to that in [12] to recover the silhouette.

2.2.1 Region Voting

This voting scheme is based upon the results yield by a classifier whose output is dependant on the foreground and background information given by the silhouettes at the i th view for each of the regions $\mathcal{R}_{\hat{p}}^i$. If a region is totally in the background, all the pixels in $\mathcal{R}_{\hat{p}}^{i+1}$ are automatically awarded a background vote. Likewise, if $\mathcal{R}_{\hat{p}}^i$ is totally inside the silhouette for the i th frame, it votes for its comprising pixels being foreground ones in the $i+1$ view. However, if the region $\mathcal{R}_{\hat{p}}^i$ covers both, foreground and background pixels, then we make use of a classifier based upon discriminant analysis to recover the region votes.

To commence, consider the pixel $p^i \in \mathcal{R}_{\hat{p}}^i$. We denote $\mathcal{V}(p^{i+1}, \mathcal{R}_{\hat{p}}^i)^F$ the votes for the pixel p^{i+1} to be considered as foreground. Conversely, the background votes for the pixel are given by $\mathcal{V}(p^{i+1}, \mathcal{R}_{\hat{p}}^i)^B$. Note that, given the silhouette at frame indexed i , the pixels in the region can be classified as foreground or background. That is, the pixels inside the silhouette are considered to be foreground and those outside the occluding contour describe the background. Hence, we can label the foreground and background pixels as members of the sets Ω_F and Ω_B , respectively.

With these ingredients, it is a straightforward task to recover a cut-off value for the pixels $p^i \in \mathcal{R}_{\hat{p}}^i$ making use of discriminant analysis [10]. At this point, it is important to note that, since we aim at recovering silhouettes from thermal imagery, we will focus our development in the use of a cut-off value based upon pixel temperatures. Nonetheless, the development presented here can

be extended, in a straightforward manner, to colour spaces. Once the temperature cut-off value is at hand, it can be used to recover the votes corresponding to $p^{i+1} \in \mathcal{R}_{\hat{p}}^{i+1}$. To do this, we commence by computing the mean and variance for the foreground and background classes in $\mathcal{R}_{\hat{p}}^i$. For the two classes, the variances are given by

$$\sigma_F^2 = \frac{1}{|\Omega_F|} \sum_{p^i \in \Omega_F} (p^i - \mu_F)^2 \quad (6)$$

$$\sigma_B^2 = \frac{1}{|\Omega_B|} \sum_{p^i \in \Omega_B} (p^i - \mu_B)^2 \quad (7)$$

where μ_F and μ_B are the mean foreground and background pixel class levels, respectively.

The optimal cut-off value $r_{\hat{p}}^i$ for the pixels in the region $\mathcal{R}_{\hat{p}}^i$ is given by that which maximizes Fisher's linear discriminant [13] separability measure given by

$$\lambda = \sigma_b^2 / \sigma_w^2 \quad (8)$$

where σ_b , σ_w are between and within class variances given by

$$\sigma_w^2 = \omega_B \sigma_B^2 + \omega_F \sigma_F^2 \quad (9)$$

$$\sigma_b^2 = \omega_B \omega_F (\mu_B - \mu_F)^2 \quad (10)$$

and ω_F , ω_B are real-valued class weights.

To take our analysis further, we note that the maximum of λ is given by $\omega^* = \omega_F = \omega_B$, where ω^* is the optimum value of the weights, which can be computed making use of the expression

$$\omega^* = \frac{\mu_B - \mu_F}{\sigma_B^2 + \sigma_F^2} \quad (11)$$

Moreover, making use of ω^* , it can be shown that the optimum cut-off value $r_{\hat{p}}^i$ is given by

$$r_{\hat{p}}^i = \{k | (\omega^*)^2 = \omega_k (1 - \omega_k)\} > 0 \quad (12)$$

where ω_k is a real-valued function of the pixel-temperature level k defined as follows

$$\omega_k = \frac{1}{|\Omega_k|} \sum_{p^i \in \Omega_k} p^i \quad (13)$$

and Ω_k is the set of pixels in $\mathcal{R}_{\hat{p}}^i$ whose level is less or equal than k , i.e. $\Omega_k = \{p^i | p^i \in \mathcal{R}_{\hat{p}}^i \wedge p^i \leq k\}$. Thus, in practice, we can recover $r_{\hat{p}}^i$ making use of sequential search governed by the condition in Equation 12.

Once the value of $r_{\hat{p}}^i$ for each of the image regions $\mathcal{R}_{\hat{p}}^i$ is at hand, the pixel p^{i+1} receives a vote $\mathcal{V}(p^{i+1}, \mathcal{R}_{\hat{p}}^i)$ from each region $\mathcal{R}_{\hat{p}}^i$ comprising the pixel p^i . A foreground vote is cast if the pixel p^{i+1} is classified as foreground based upon the cut-off

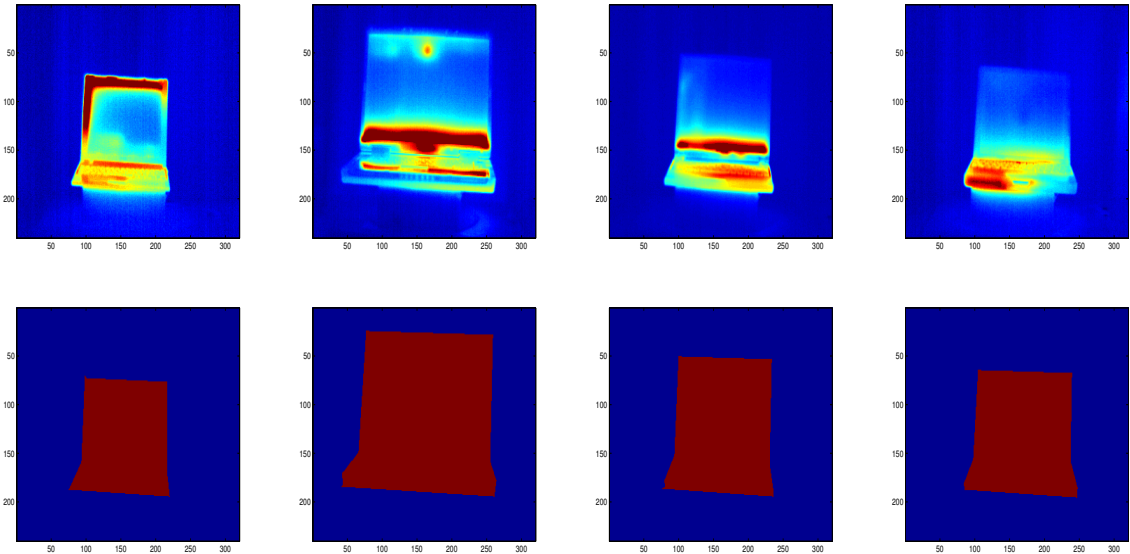


Figure 1: Top row: First frame for the image sequences under study; Bottom row: Corresponding input silhouettes.

value r_p^i for the region \mathcal{R}_p^i . A foreground vote is cast otherwise. As a result, the votes are dependant on the classification given to the temperature value of the pixel at frame $i + 1$ as yield by the classifier trained using the region information on the i th view.

2.2.2 Silhouette Recovery

Once the pixel votes are at hand, we can make use of the method in [12] to separate the foreground from the background and, hence, recover the object’s silhouette. This is possible due to the fact that the foreground-vote histogram for the view indexed $i + 1$ can be normalized and regarded as a probability distribution which can be dichotomized in an unsupervised fashion.

This separation in two classes is, in fact, the result of the assumption, made *a priori* that the pixels in the image belong to either the background or the foreground classes. Recall that the algorithm in [12] is based upon discriminant analysis. Our choice of threshold criterion responds to the intuition that the best separation of classes, in terms of foreground votes, would yield the best silhouette possible.

3 Experiments

In this section, we turn our attention to the experimental evaluation of our silhouette recovery method. To this end, we have used the thermal image-sequences corresponding to four laptops. For the sake of simplicity, we have acquired the

imagery making use of a turntable, on which the laptop under study has been rotated and translated. Thus, our image sequences show the views for the objects under study from 0° to 360° in 10° increments. The translation was of ± 10 cm perpendicular to the camera axis in steps of 1cm between views.

In Figure 1, we show the first view for each of the four laptops under study and the manually extracted contour provided at input. From the panels, it can be noted that the variation of temperature across the object is considerable. Furthermore, some regions, such as the screen and areas of the keyboard have a temperature value which is comparable to that of the background.

In the top row of Figure 2, we show example results for the 12th view in our test sequences. The middle row show the foreground-vote maps, i.e. the normalized magnitude of the foreground votes sets for each of the pixel locations on the image. The silhouette recovered by our algorithm are shown in the bottom row. From the figure, we can conclude that, despite the small temperature difference between the foreground and some foreground areas, the recovered silhouettes are in close accordance with the occluding contour of the objects. For instance, the screen of the fourth laptop, i.e. the laptop shown in the fourth column, is barely distinguishable from the background. Nonetheless, the foreground-vote maps show a clear distinctness between the object and the background.

Finally, we turn our attention to a more quantitative evaluation of the results yield by our algo-

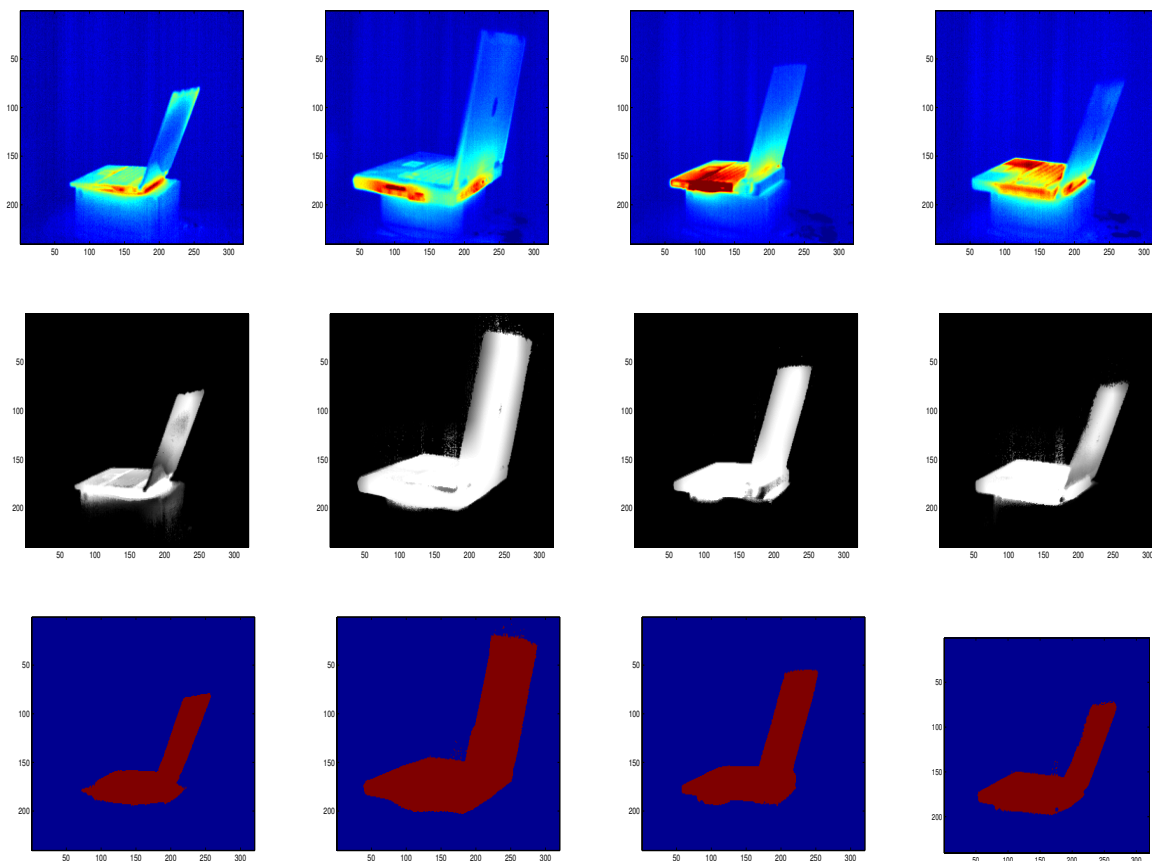


Figure 2: Top row: input views; Middle row: foreground-vote maps for the views in the top row; Bottom row: recovered silhouettes

Table 1: Percentage of Misassigned Foreground Pixels for each of the objects under study

Object under study	mean \pm standard deviation
Laptop 1	0.0208 \pm 0.0049
Laptop 2	0.0257 \pm 0.0049
Laptop 3	0.0236 \pm 0.0049
Laptop 4	0.0219 \pm 0.0050

rithm. To this end, we have used ground-truth silhouette data and, for each of the objects under study, we have computed the average percentage of misassigned foreground pixels across the 36 views comprising each image sequence. In Table 1, we list the mean and standard deviation for the percentage of misassigned pixels as a function of object index. From the quantitative results shown we conclude that the silhouette recovered by our algorithm is in close accordance with the ground truth. Furthermore, the algorithm can cope well with the recovery of silhouettes for objects whose surface temperature is comparable to that of the background.

4 Conclusions

We have presented a method for extracting silhouettes from uncalibrated thermal imagery. Our method is semi-supervised in nature and propagates the information provided by the user, in the form of a single silhouette, across the views under study. We have shown how this can be effected making use discriminant analysis techniques. We have shown results on real-world imagery.

5 Acknowledgements

National ICT Australia is funded by the Australian Governments Backing Australia’s Ability initiative, in part through the Australian Research Council.

References

- [1] P. Sander, X. Gu, S. Gortier, H. Hoppe, and J. Snyder, “Silhouette clipping,” in *SIGGRAPH 2000*, pp. 327–334, 2000.
- [2] G. Cheung, S. Baker, and T. Kanade, “Shape-from-silhouette of articulated objects and its

- use for human body kinematics estimation and motion capture,” in *IEEE International Conference of Computer Vision*, pp. 77–84, 2003.
- [3] C. Hernández Esteban and F. Schmitt, “Silhouette and stereo fusion for 3d object modeling,” *Computer Vision and Image Understanding*, vol. 96, no. 3, pp. 367–392, 2004.
- [4] R. Raskar and M. F. Cohen, “Image precision silhouette edges,” in *ACM Symposium on Interactive 3D Graphics*, pp. 135–140, 1999.
- [5] L. Kettner and E. Welzl, “Contour edge analysis for polyhedron projections,” in *Geometric Modeling: Theory and Practice*, pp. 379–394, 1997.
- [6] S. Sullivan and J. Ponce, “Automatic model construction and pose estimation from photographs using triangular splines,” *IEEE Pattern Analysis and Machine Intelligence*, vol. 20, pp. 1091–1097, October 1998.
- [7] D. Snow, P. Viola, and R. Zabih, “Exact voxel occupancy with graph cuts,” in *In Proc. Computer Vision and Pattern Recognition Conf.*, vol. 1, pp. 345–352, 2000.
- [8] J. Zheng, “Acquiring 3-d models from a sequence of contours,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 16, no. 2, pp. 163–178, 1994.
- [9] G. Zeng and L. Quan, “Silhouette extraction from multiple images of an unknown background,” *ACCV04*, 2004.
- [10] K. Fukunaga, *Introduction to statistical pattern recognition*. Academic Press, Inc, 1990.
- [11] R. Hartley and A. Zisserman, *Multiple view geometry in Computer Vision*. Cambridge University Press, 2000.
- [12] N. Otsu, “A thresholding selection method from gray-level histograms,” *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
- [13] R. A. Fisher, “The use of multiple measurements in taxonomic problems,” *Annals of Eugenics*, vol. 7, pp. 179–188, 1936.

A simple and efficient eye detection method in color images

D. Sidibe, P. Montesinos, S. Janaqi

LGI2P/EMA - Ales School of Mines

Parc scientifique G. Besse, 30035 Nimes Cedex 1, France

Email:{Desire.Sidibe, Philippe.Montesinos, Stefan.Janaqi}@ema.fr

Abstract

In this paper we propose a simple and efficient eye detection method for face detection tasks in color images. The algorithm first detects face regions in the image using a skin color model in the normalized RGB color space. Then, eye candidates are extracted within these face regions. Finally, using the anthropological characteristics of human eyes, the pairs of eye regions are selected. The proposed method is simple and fast since it needs no template matching step for face verification. It is robust because it can deal with rotation. Experimental results clearly show the validity of our approach. A correct eye detection rate of 98.4% is achieved using a subset of the AR face database.

Keywords: Eye detection, Skin detection, Skin color model, Face detection

1 Introduction

Automatic human face analysis and recognition has received significant attention during the past decades, due to the emergence of many potential applications such as person identification, video surveillance and human computer interface. An automatic face recognition usually begins with the detection of face pattern, and then proceeds to normalize the face images using information about the location and appearance of facial features such as eyes and mouth [1], [2]. Therefore, detecting faces and facial features is a crucial step. Many methods for solving the face detection problem have been proposed in the literature (see [3] for a more detailed review) and most of them can be put into a two-stage framework [4]. The first stage focuses attention to face candidates, i.e. regions that may contain a face are marked. In the second stage, the face candidates are sent to a "face verifier", which will decide whether the candidates are real faces or not. Different methods put emphasis on one or other of these stages.

Eyes can be considered the most salient and stable features in a human face in comparison with other facial features. Therefore, extraction of eyes is often a crucial step in many face detection algorithms [5], [6]. A recent review on eye detection techniques can be found in [7]. The main classical methods include the template matching method, eigenspace method and Hough transform method [8], [9]. Besides these three classical methods, many other image-based eye detection techniques have been proposed recently. Han et al. [5] use mor-

phological operations to locate eye-analogue pixels in the input image. Then a labeling process is executed to generate eye-analogue segments which are used as guides to search for potential face regions. Finally a trained backpropagation neural network is used to identify faces and their locations. Similar ideas are used by Wu and Zhou [4]. They employ size and intensity information to find eye-analogue segments from gray scale image, and exploit geometrical relationship to filter out the possible eye-analogue pairs. They also use a template matching approach for face candidates verification. Huang and Wechsler [10] use genetic algorithms to evolve some finite state automata to discover the most likely eye locations. Then optimal features are selected and a decision tree is built to classify whether the most salient locations identified earlier where eyes. Kawaguchi and Rizon [11] use intensity and edges information to locate the iris. The main techniques they use are template matching, separability filter and Hough transform. Song et al. [7] use similar ideas to detect eyes. An improvement of their work is the extraction of binary edge images based on multi-resolution wavelet transform.

In this paper, a simple and robust eye detection method in color images is presented. The proposed method strongly depends on a good face region selector. A skin color model is used to select face regions. Then eyes are directly detected within these regions based on anthropological characteristics of human eyes. The method is simple since it needs no training examples of eyes or faces, and no face verification step. The remainder of the paper

is organized as follows. The face region detection is described in Section 2. The eye detection algorithm is addressed in Section 3. Some experimental results showing the validity of the method, are given in Section 4. Finally, concluding remarks are given in Section 5.

2 Face region detection

Human skin color is a very efficient feature for face detection. Although different people may have different skin color, several studies have shown that the major difference lies largely between their intensity rather than their chrominance [12], [13]. Many different color spaces have been employed. Among them one finds: RGB, normalized RGB, HSI, HSV, YCbCr, YES, YUV, CIE Lab [6]. Terrillon et al. [14] have shown that the tint-saturation-luma (TSL) space and the normalized RGB space provide best results for Gaussian models. But we can notice, following Albiol et al. [15], that if an optimal skin detector is designed for every color space, then their performance will be same. For that reason, we adopt the normalized RGB color space since it is simple and we model the skin distribution by a single Gaussian.

2.1 Skin color modeling

Skin color distribution can be modelled by an elliptical Gaussian probability density function (pdf), defined as:

$$f(c|skin) = \frac{1}{2\pi|\Sigma_s|^{1/2}} e^{-\frac{1}{2}(c-\mu_s)^T \Sigma_s^{-1}(c-\mu_s)} \quad (1)$$

where c is a color vector and (μ_s, Σ_s) are the distribution parameters. These parameters are estimated from a training sample. We used a set of 1,158,620 skin pixels, manually selected from about 100 Internet images. The images are chosen in order to represent people belonging to several ethnic groups, and a wide range of illumination conditions.

A more sophisticated model, a mixture model, is often used in the literature [16], [14]. It is a generalization of the single Gaussian and the pdf in this case is the sum of several single Gaussians. The reason why we choose a single Gaussian model is that our experiments have shown that the performance of mixture models exceeds single model's performance only when a high true positive rate is needed (more than 80%). The same observation have been given by Caetano et al. in [17].

2.2 Skin detection

Once the parameters of skin color distribution in the normalized RGB color space are obtained from

the training sample, we use the Mahalanobis distance from the color vector c to mean vector μ_s , given the covariance matrix Σ_s to measure how "skin like" the color c is:

$$\lambda_s(c) = (c - \mu_s)^T \Sigma_s^{-1} (c - \mu_s) \quad (2)$$

Given an input image, for each pixel x , $x = (r, g)$ in the normalized RGB color space, x is considered a skin pixel if $\lambda_s(x) \leq t$. In our experiments, the threshold value t was chosen to obtain a true positive rate of 80%, while ensuring a false positive rate less than 15%. An example of skin detection result using an image from the AR database is shown in figure 1.



Figure 1: From left to right: original image, skin region detected.

3 Eye detection

In [4] and [5], eyes are detected based on the assumption that they are darker than other part of the face. Han et al. [5] use morphological operations to locate eye-analogue segments, while Wu and Zhou [4] find eye-analogue segments searching small patches in the input image that are roughly as large as an eye and are darker than their neighborhoods. In these methods, eye-analogue segments are found in the entire image resulting in a high number of possible pairs to check. On the contrary, in the proposed method, we directly search for eye-analogue segments within the face region. We consider as potential eye regions, the non-skin regions within face region. Obviously, eyes should be within a face region and eyes are not detected as skin by the skin detector. The same ideas are used by Hsu et al. [6]. Therefore, we have to find eye-analogue pairs among a reduced number of potential eye regions (see figure 2).

An ellipse is fitted to each potential eye region using a connected component analysis. Let R_k be a potential eye region and (x_k, y_k) its centroid. Then R_k , reduced to an ellipse, defines a_k , b_k and θ_k which are, respectively, the length of the major axis, the length of the minor axis and the orientation of the major axis of the ellipse.

Finally, a pair of potential eye regions is considered as eyes if it satisfies some constraints based on

anthropological characteristics of human eyes. Let R_i and R_j be two potential eye regions. Then (R_i, R_j) corresponds to a pair of eyes if the following equations are satisfied:

- $$\begin{cases} 1 < \frac{a_i}{b_i} < 3 \\ 1 < \frac{a_j}{b_j} < 3 \end{cases} \quad (3)$$

- $$|\theta_i - \theta_j| < 20^\circ \quad (4)$$

- $$\frac{a_i + a_j}{2} < d_{ij} < 3 \frac{a_i + a_j}{2} \quad (5)$$

The parameters in equation (3) and equation (5) are chosen according to the fact that for human eyes, if we denote by w_e and h_e respectively the width and the height of an eye, the average value for w/h is 2 and averagely $d_{ij} = 2w_e$ [18]. Equation (4) is based on the fact that the two major axis should have the same orientation. A final constraint is the alignment of the two major axis, i.e. for two eye regions they belong to the same line.



Figure 2: From left to right: skin region detected, potential eye regions.

Using these rules, the algorithm sometimes detects not only eyes, but also eyebrows. To discard regions corresponding to eyebrows, we use the fact that the center part of an eye region is darker than other parts. Then a simple histogram analysis of the region is done for selecting eye regions since an eye region should exhibit two peaks while an eyebrow region shows only one.

4 Experimental results

We made different experiments to evaluate the performance of the proposed algorithm. Firstly, we used the AR face database [19] to compare our results with those described by Kawaguchi and Rizon [11], and Song et al. [7]. This database contains color images of frontal view faces with different facial expressions, illumination condition and occlusions. For a direct comparison, we used the same subset of the database employed in [11] and

[7]. This subset, named AR-63, contains 63 images of 21 people (12 men and 9 women) without spectacles stored in the first CD ROM. The images in AR-63 show three expressions (neutral, smile and anger) and have natural illumination condition.

Secondly, we used some images gathered from Internet for testing the robustness of the method against complex background, varying illumination condition and rotation.

4.1 Evaluation criterion

A commonly used criterion for the performance evaluation of an eye detection method is *the relative error* introduced by Jesorsky et al. [20]. It is defined by:

$$err = \frac{\max(d_l, d_r)}{d_{lr}} \quad (6)$$

where d_l is the left eye disparity, i.e. the distance between the manually detected eye position and the automatically detected position, d_r is the right eye disparity, and d_{lr} is the Euclidean distance between the manually detected left and right eye positions. In [4], the detection is considered to be correct if $err < 0.25$. Song et al. [7] defined an other criterion. They considered the detection to be successful if:

$$\max(d_l, d_r) < \alpha r \quad (7)$$

where r is the radius of an iris and α is a scalar factor. Considering that the radius of an iris is about $\frac{1}{4}$ of an eye width, one can see that the criterion of equation (6) is equivalent that of equation (7) with $\alpha = 2$.

4.2 Results and discussion

Using the subset AR-63, the proposed method achieves a success rate of 100% based on the criterion defined in equation (6), and a success rate of 98.4% (one failed image) based on the criterion defined in equation (7) for $\alpha = 1$. Some detection results are shown in figure 3 where an eye is depicted by a small white cross.

Comparing the proposed method with those described by Kawaguchi and Rizon [11], and Song et al. [7] using the same set of data, we see that the performance of our method is equivalent to that of the method of Song et al. (98.4% of correct detection), and both methods obtain slightly better results than the method of Kawaguchi and Rizon (96.8% of correct detection). The methods in [11] and [7] can deal with gray scale images but they need to detect the reflected light dots as a cue for eye localization. One main advantage of our method is that we obtain very precise eye

localization without the detection of the reflected light dots.



Figure 3: Examples of detected eyes by the proposed method using the subset AR-63.

Figure 4 and figure 5 show some detection results which demonstrate the robustness of the method against rotation and illumination condition. The skin detector is robust enough to deal with different illumination conditions and the algorithm is rotation invariant because we made no assumption about the face orientation for detecting eyes.



Figure 4: Other examples of eye detection.

One can also notice, figure 5, that the method can be successful when multiple faces are present. Nevertheless, there are some eyes which are not detected in that case. In particular, closed eyes can not be detected.

The most related work to ours is the work of Hsu et al. [6]. They base their face detection algorithm on a robust skin detector too. Then they extract eyes and mouth as facial features by constructing eye and mouth maps based on the luminance and the chrominance components of the image. Finally, they form an eye-mouth triangle for all possible



Figure 5: Example of multiple faces detection.

combinations of the eye candidates and one mouth candidate. Each eye-mouth triangle is verified using a score and the Hough transform. While this method gives good results and may be more robust than ours, we have found that mouth is a less stable feature than eyes since we do not use an explicit mouth or eye map. Moreover, using simple rules to detect eyes, the proposed method is faster than the one described in [6]. The average execution time, given in [6], for processing an image (size 640 x 480) on a 1.7 GHz CPU is 24.71 s. The average time for processing an image (size 768 x 576) on a 3 GHz CPU with our method is 3.8 s.

5 Conclusion

In this paper a simple and efficient eye detection method for detecting faces in color images is proposed. It is based on a robust skin region detector which provides face candidates. Then using some simple rules derived from anthropological characteristics, eyes are selected within face regions. The procedure is robust enough to avoid a face verification system and it achieves a successful rate of 98.4% on a subset of the AR face database. It can also detect eyes in case of rotation and in the presence of multiple faces.

The speed of the method and the robustness to rotation would be very useful for real-time applications. However, experimental results show that the method may fail if one or both eyes are closed and if the face is viewed in profile.

Further improvements can be done for the detection of multiple faces with different orientations and sizes. A multi-scale approach can be used for that.

Acknowledgements

We would like to thank the anonymous reviewers for their useful comments and suggestions.

References

- [1] S. G. Kong, J. Heo, B. R. Abidi, J. Paik, and M. A. Abidi, "Recent advances in visual and infrared face recognition: a review," *Computer Vision and Image Understanding*, vol. 97, pp. 103–135, 2005.
- [2] W. Zhao, R. Chellappa, A. Ronsenfeld, and P. J. Phillips, "Face recognition: A literature survey," *ACM Computing Surveys*, pp. 399–458, 2003.
- [3] M.-H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 34–58, january 2002.
- [4] J. Wu and Z. H. Zhou, "Efficient face candidates selector for face detection," *Pattern Recognition*, vol. 36, pp. 1175–1186, 2003.
- [5] C. C. Han, H. Y. M. Liao, G. J. Yu, and L. H. Chen, "Fast face detection via morphology-based pre-processing," *Pattern Recognition*, vol. 33, pp. 1701–1712, 2000.
- [6] R. Hsu, M. Abdel-Mottaleb, and A. K. Jain, "Face detecting in color images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 696–706, may 2002.
- [7] J. Song, Z. Chi, and J. Liu, "A robust eye detection method using combined binary edge and intensity information," *Pattern Recognition*, vol. 39, pp. 1110–1125, 2006.
- [8] R. Brunelli and T. Poggio, "Face recognition: features versus templates," *IEEE Trans. on Pattern Analysis and Machine Intel.*, vol. 15, no. 10, pp. 1042–1052, 1993.
- [9] A. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition," in *IEEE Proc. of Int. Conf. on CVPR*, pp. 84–91, Seattle, Washington, 1994.
- [10] J. Huang and H. Wechsler, "Visual routines for eye location using learning and evolution," *IEEE Trans. on Evol. Comput.*, vol. 4, no. 1, pp. 73–82, 2000.
- [11] T. Kawaguchi and M. Rizon, "Iris detection using intensity and edge information," *Pattern Recognition*, vol. 36, pp. 549–562, 2003.
- [12] H. Graf, T. Chen, E. Petajan, and E. Cosatto, "Locating faces and facial parts," in *Proc First Int'l Workshop Automatic Face and Res-ture Recognition*, pp. 41–46, 1995.
- [13] M. J. Jones and J. M. Rehg, "Statistical color models with application to skin detection," *International Journal of Computer Vision*, vol. 46, no. 1, pp. 81–96, 2002.
- [14] J. Terrillon, M. N. Shirazi, H. Fukamachi, and S. Akamatsu, "Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images," in *Proc. IEEE Int'l Conf. on Face and Gesture Recognition*, pp. 54–61, 2000.
- [15] A. Albiol, L. Torres, and E. J. Delp, "Optimum color spaces for skin detection," in *ICIP*, pp. 122–124, 2001.
- [16] H. Greenspan, J. Goldberger, and I. Eshet, "Mixture model for face-color modeling and segmentation," *Pattern Recognition Letters*, vol. 22, pp. 1525–1536, 2001.
- [17] T. S. Caetano, S. D. Olabarriaga, and D. A. C. Barone, "Do mixture models in chromaticity space improve skin detection?," *Pattern Recognition*, vol. 36, pp. 3019–3021, 2003.
- [18] A. M. Alattar and S. A. Rajala, "Facial features localization in front view head and shoulders images," in *IEEE Proc. of ICASSP*, vol. 6, pp. 3557–3560, 1999.
- [19] A. M. Martinez and R. Benavente, "The AR face database," Tech. Rep. 24, CVC, june 1998.
- [20] O. Jesorsky, K. J. Kirchberg, and R. W. Frischholz, "Robust face detection using the hausdorff distance," in *Proc. of the Third Int'l Conf. on Audio and Video-based Biometric Person Authentication*, pp. 90–95, 2001.

Access Control with Session Based Face Tracking

Amadeus Rainbow and Richard Green

University of Canterbury, Dept. Computer Science and Software Engineering.

Email: richard.green@canterbury.ac.nz

Abstract

This paper discusses and presents computer vision research using an eigenface algorithm for session based face tracking for access control on computer screens in public areas. In a typical office setup, the algorithm performed well by locking the screen when a second face was present or as soon as the authenticated user left the view of the terminal. The results culminated in a false negative and locked screen 2% of the time, supporting the algorithmic approach to session based security.

Keywords: access control, session based, face tracking, face recognition, eigenface

1 Introduction

Everyday computers display critical and private information about individuals and companies. Computer systems use access control [1], consisting of identification, authentication and restrictions on users to keep this information secure. However, the system has no control over access to a computer terminal once a user has been authenticated. It is the user's responsibility to hide sensitive information from unauthorised persons and to lock the screen when leaving a terminal unattended [2]. The advent of open plan offices, customer friendly service desks, Netcafes and wireless local area networks combined with trusted users, such as secretaries or customer service representatives being undereducated in computer security or lacking secure habits, have made it easy for unauthorised individuals to view sensitive information on other's computer screens [3]. A low cost solution is to use web cameras with face recognition [4] to track the user while they are logged in. According to Zhao, Chellappa, Phillips and Rosenfeld's literature survey [5], face tracking and recognition has evolved in the past 30 years to a stage where it is reliable enough for commercial, real-time applications. However, They also found that the recognition of faces in outdoor environments with changes in illumination and pose remains a largely unsolved problem. Zhang, Yan and Lades, who compared eigenface, authentication and classification neural nets, and elastic matching [6], found that the eigenface algorithm was the easiest to implement and the least computationally taxing on hardware. Tan and See, who compared normalised cross-correlation, gradient image, relative gradient image feature and eigenface [7], found that even though the eigenface method has a lower tolerance for severe illu-

mination variations, yaw angle variations of the head and facial expressions variations than their other tree face recognition approaches, the superior speed of eigenfaces made it the most appropriate recognition algorithm for real-time applications.

2 Implementation

The prototype system uses a 1.3 mega pixel Logitech web camera [8] mounted on the monitor of the workstation. When the prototype program starts, five training pictures of the authorised user are taken. These pictures are then added to a library of eighteen pictures that are used to create twelve eigenfaces [9], similar to figure 1. The program then takes a frame from the USB web camera finds a face and compares it to the eigenfaces, every second. When the face is recognised as the user's, the screen remains unlocked but when the user's face is no longer present or a second face is detected, the screen is locked.

2.1 Face Search

The prototype program uses OpenCV's cvHaarDetectObjects [10] function which scans by sliding a Haar classifier cascade across the image several times, at different scales. During every pass, the function applies the classifiers to overlapping regions in the image. The function also applies Canny pruning heuristics [11] to reduce the number of regions needing analysis. Then the function returns a sequence of average rectangles for each large enough group of regions that passed the classifier cascade. When no rectangles are returned, the program assumes there are no faces. If more than one rectangle is returned, the program assumes there are other faces apart

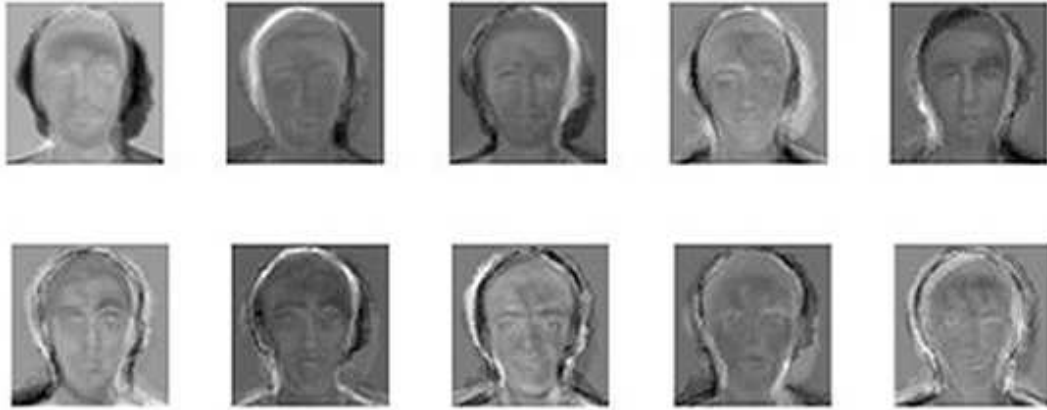


Figure 1: Graphical representation of 10 Eigenfaces.

from the user's present. When one rectangle is returned, the program cuts that region out of the frame and passes it to the recognition functions as the "input image".

2.2 Eigenface Creation

Every image in a set of M original face images, is transformed into a one dimensional vector of size N and placed into the set $S = \{\Gamma_1, \Gamma_2, \Gamma_3, \dots, \Gamma_M\}$. After obtaining the mean image Ψ , shown in figure 2,



Figure 2: Graphical representation of an average face, calculated from a set of portraits.

$$\Psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n \quad (1)$$

the difference Φ between the input image and the mean image is found.

$$\Phi_i = \Gamma_i - \Psi \quad (2)$$

Next a set of M orthonormal vectors, u_n , which best describes the distribution of the data is calculated. The k^{th} vector, u_k , is chosen such that

$$\lambda_k = \frac{1}{M} \sum_{n=1}^M (u_k^T \Phi_n)^2 \quad (3)$$

is a maximum, subject to

$$u_l^T u_k = \delta_{lk} = \begin{cases} 1 & (l = k) \\ 0 & (l \neq k) \end{cases} \quad (4)$$

The vectors u_k and scalars λ_k are the eigenvectors and eigenvalues of the covariance matrix C ,

$$\begin{aligned} C &= \frac{1}{M} \sum_{n=1}^M \Phi_n \Phi_n^T \\ &= AA^T \end{aligned} \quad (5)$$

where $A = [\Phi_1, \Phi_2, \Phi_3, \dots, \Phi_n]$ and A^T is

$$L_{mn} = \Phi_m^T \Phi_n \quad (6)$$

once the eigenvectors v_l and u_l are found using

$$u_l = \sum_{k=1}^M v_{lk} \Phi_k \quad (7)$$

for $l = 1, 2, \dots, M$.

2.3 Recognition

The input image is compared to the mean image and their difference is multiplied with each eigenvector of the L matrix.

$$\omega_k = u_k^T (\Gamma - \Psi) \quad (8)$$

for $k = 1, 2, \dots, M$. Each value represents a weight and is saved in a vector $\Omega^T = [\omega_1, \omega_2, \dots, \omega_M]$. The Euclidean distance is minimised,

$$\epsilon_k = \|\omega - \omega_k\|^2 \quad (9)$$

to determine which face class, or training picture, provides the closest match to the input image. For the closest match, the distance ϵ_k is below an established threshold θ_ϵ . If the difference is above that threshold, no match is found.

3 Experiments and Results

To make testing easier the prototype system displays two windows, one shows what the web camera “sees” and the other the result, which is the closest match to the face found in the current frame. Instead of actually locking the screen the program displays “Screen Locked” with a one line description as to why, in the results window. The lighting conditions and trained user’s face were kept constant throughout the following tests.

3.1 Lighting

The prototype could no longer recognise the user’s face when, the users face was trained in a high lighting condition and the room was changed to a low lighting condition, by turning off the lights. The same thing happened when the user’s face was trained in a low lighting condition and the room was changed to a high lighting condition.

3.2 Distance

After experimentation it was found that for the program to recognise a trained face or a second face reliably, the face had to occupy at least 900 pixels, which in the Logitech web camera’s [8] case implied a maximum rang of two metres.

3.3 Printed Face

After experimentation it was also found that the program accepted a printout of a high-resolution picture of a face as a user. And when a user already existed the program accepted the printout as a second face.

3.4 Consistency

The prototype returns one of four cases: current face recognized as the user, no match found for the current face, two or more faces present or no current face found. Each case had its own test with one hundred input frames. Each test recorded what the prototype returned to make table 1.

3.4.1 Users Face

A real user who moved around as much as someone working at a computer workstation, was seated in front of the computer to test that the program would recognize the user and keep the screen unlocked.

3.4.2 Two of More Faces

After the program was trained with the user’s face, a printout of a high-resolution picture of a face was held next to the user to test that the program found two or more faces and locked the screen.

3.4.3 Other Face

After the program was trained with the user’s face, a printout of a high-resolution picture of a face was held in front of the web camera to test that the program found a face, did not recognise it as the user’s face and locked the screen.

3.4.4 No Faces

After the program was trained with the user’s face, the user left the web camera’s field of vision to test that the program did not find any faces and locked the screen.

3.4.5 Results

Surprisingly, under normal conditions with constant lighting, the screen locked every time the user left the view of the web camera or a second face appeared within two meters of the web camera. The screen locking on the user because of false negatives; not matching the user’s face or not finding the user’s face, only occurred 2% of the time. This is shown in table 1.

4 Conclusion and Future Work

With a 100% screen locking rate, in a typical office setup, this is an ideal session based security system for workstations that have multiple users, over the course of the day. In this setup lighting is constant and not a problem, however the program could be made more robust by retraining the user’s face every time a large lighting change is registered by the web camera. The program could also average the current state with the previous state, to reduce false negatives. In future, the system could use stereo cameras, to add depth perception and combat the 2% failure rate for false negatives, the acceptance of a picture as face and to make a well defined area where faces can and can not see the

Table 1: Results from Consistency Tests

	User's Face	Two Faces	Other Face	No Faces
Recognised User	98%	0%	0%	0%
No Match Found	1%	1%	99%	1%
Found Two or More Faces	0%	99%	0%	0%
No Face Found	1%	0%	1%	99%
Locked Screen	2%	100%	100%	100%

computer screen. In conclusion, a web camera and a session based face tracking approach is cost effective and robust enough to use as access control on reasonably public computer screens.

References

- [1] R. S. Sandhu and P. Samarati, "Access control: principle and practice," *Communications Magazine, IEEE*, vol. 32, no. 9, pp. 40–48, 1994.
- [2] M. A. Sasse, S. Brostoff, and D. Weirich, "Transforming the weakest link a human/computer interaction approach to usable and effective security," *BT Technology Journal*, vol. 19, no. 3, pp. 122–131, 2001.
- [3] G. Dhillon and J. Backhouse, "Technical opinion: Information system security management in the new millennium," *Communications of the ACM*, vol. 43, no. 7, pp. 125–128, 2000.
- [4] C. Mallauran, J. Dugelay, F. Perronnin, and C. Garcia, "Online face detection and user authentication," in *Proceedings of the 13th annual ACM international conference on Multimedia*, pp. 219–220, 2005.
- [5] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Computing Surveys (CSUR)*, vol. 35, no. 4, pp. 399–458, 2003.
- [6] J. Zhang, Y. Yan, and M. Lades, "Face Recognition: Eigenface, Elastic Matching, and Neural Nets," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1423–1435, 1997.
- [7] K. Y. Tan and A. K. B. See, "Comparison and implementation of various methods in facial recognition technology," *GVIP Journal*, vol. 5, no. 9, pp. 11–19, 2005.
- [8] Logitech, "Quickcam pro5000." <http://www.logitech.co.nz>, visited on 21/9/2006.
- [9] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference on*, pp. 586–591, Jun 1991.
- [10] G. Bradski, A. Kaehler, and V. Pisarevsky, "Learning-based computer vision with intel's open source computer vision library," *Intel Technology Journal*, vol. 9, no. 1, pp. 118–131, 2005.
- [11] A. Wallack and J. Canny, "Efficient indexing techniques for model based sensing," *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pp. 259–266, 1994.

A New Rapid Feature Extraction Method for Computer Vision based on Moments

A. L. C. Barczak and M. J. Johnson¹

¹Institute of Information and Mathematical Sciences, Massey University, Auckland

Email:A.L.Barczak@massey.ac.nz

Abstract

In this paper we propose a new image feature extraction method for Pattern Recognition. We combine *Summed-area Tables* and *Moment Invariants* to rapidly compute geometric moments over areas (sub-windows) of the image. An approximation of a circular area allows to compute moments from areas defined by concentric circles (concentric discs). The advantages of this method are: fast feature extraction, rotation invariance and scaling invariance. The fact that the set of features are rotation invariant but not translation invariant is the key to locate objects unambiguously when a multi-resolution scan is applied to the image. To illustrate the method we computed the moments of concentric discs for a small set of images and analyse its accuracy. The experiments showed a good potential of the method to be used in real-time applications such as object detection and tracking.

Keywords: Feature Extraction, Pattern Recognition, Moment Invariants, Viola-Jones Detector.

1 Introduction

Feature extraction is the most critical stage of many Computer Vision applications such as object detection and recognition. However the computation of extra features from the images is usually too slow to be used in real-time applications. Usually one has to choose a compromise among certain characteristics such as invariability, discriminating powers, dimensionality and computational complexity of the feature extraction. Feature extraction based on Moment Invariants has the strength of keeping the same values despite certain geometric transformations of the image (such as rotation, scaling, translation etc).

Two problems arise from using features based on Moment Invariants. The first is the speed in which the image can be scanned (different sub-windows at various position and scale). The second is the limitation of the dimensions of the feature space to a small number of low order moments. This is due to the fact that higher order moments are too sensitive to noise.

In this paper we propose a new feature extraction method that combines two well known methods, namely Summed-area tables ([1]) and Moment Invariants ([2]). Our method consists in extracting geometric moments over circular areas, so several concentric discs can be examined. The moments

can be computed very rapidly at any scale and position via a set of modified Summed-area tables.

The next sections are organised as follows: a brief literature review discuss related work. We then show how to rapidly compute a set of 11 Moment Invariants using 15 Summed-area tables. Next we present the new approach to increase the feature space dimension without using higher order Moments (we limit the order to the 4th). An experimental section presents results and discussions for a simple image to analyse the invariance of the proposed features extraction method.

2 Related Work

Summed-area tables have been used to rapidly compute sums of pixels over rectangular areas ([3]). Viola and Jones used this approach to compute Haar-like features. However Haar-like features are not rotation invariant, a challenge for certain types of detection.

In the last few years there is a renewed interest for the Moment Invariants theory proposed by Hu in 1962. Flusser discussed the independence and completeness of the original Hu's set ([4]). Two of the seven moments were dependent, leaving only five to effectively be used for classification of images. Moreover, Flesser developed a method to find out the best sets of moments for higher orders. Higher order moments are known to be very sensitive to

noise, and for that reason this work is limited to extract moments up to the 4th order.

The idea of extracting features from circular areas is not new. Several methods proposed concentric discs, so different parts of the image can be analysed without losing the rotation invariance property. To cite a few examples, Arof et al. [5] used a circular neighbourhood to classify texture. Torres et al. [6] proposed a feature extraction method based on the number of intensity changes in pixels located at concentric discs. The centre of the circle is located on the centroid of the object, obtained via the moment of inertia of the image. Kazhdan et al. [7] used concentric discs to compute symmetry descriptors for 2D images. Mukundan [8] proposed the use of Radial Tchebichef Invariants (which are inherently computed over circular areas) for feature extraction and investigated their representation capabilities and their invariant properties. Another set of moments that are rotation invariant are the Zernike moments (for a comparative analysis see [9]).

To the best of the author's knowledge this is the first time that the moment invariants and the summed-area tables are combined to compute moments of concentric discs as proposed in this work.

3 Rapid computation of Moment Invariants

In this section we derive the equations to use Summed-area tables to compute 11 moments over a rectangular area.

3.1 Summed-area Tables

Summed-area tables [1] can be defined as matrices in which each element contains the sum of all the pixels that belong to the upper left parts of the original image. Given an image $i(x_i, y_i)$, the Summed-area Table $I(x, y)$ is:

$$I(x, y) = \sum_{x \leq x_i} \sum_{y \leq y_i} i(x_i, y_i) \quad (1)$$

Once a Summed-area Table is created for a certain image, the sum of rectangular areas over the image can be computed with 4 look-ups. Due to this characteristic, a recursive algorithm for creating the table can be based on the following equation:

$$I(x, y) = I(x - 1, y) + I(x, y - 1) - I(x - 1, y - 1) + i(x, y) \quad (2)$$

Where $I(x, y)$ is the integral image element and $i(x, y)$ is the image element for the point (x, y) . In

order to avoid negative indexes the integral image is padded with zeros in the first row and column.

Next we show how to deduce the equations for computing Moment Invariants at any position and scale within an image using Summed-area Tables. For clarity we present here the equations in the form that they are usually presented in text books for image processing (e.g. [10] and [11]).

3.2 Moment Invariants up to the 4th order

Given a digital image $i(x, y)$, the 2D moment of order $(p + q)$ is:

$$m_{pq} = \sum_x \sum_y x^p y^q i(x, y) \quad (3)$$

For any order $(p+q)$, each element can be pre-computed by multiplying the pixel value by its position. It is trivial to create Summed-area tables for 2D moments of any order.

2D moments are non-invariant but they are the basis for Hu's equations. If the values \bar{x} and \bar{y} are given by:

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad \text{and} \quad \bar{y} = \frac{m_{01}}{m_{00}} \quad (4)$$

A central moment is defined by:

$$\mu_{pq} = \sum_x \sum_y (\bar{x} - x)^p (\bar{y} - y)^q i(x, y) \quad (5)$$

And the normalised central moment is given by:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma} \quad (6)$$

Where $\gamma = \frac{p+q+2}{2}$

[2] proposed that seven Moment Invariants, up to the 3rd order, were to be used as the basis of image recognition systems. However subsequent work done on Moment Invariants (see thorough discussion in [4]) showed that only five of them are independent and to complete the set up to the 3rd order a new moment was needed. We adopted Flusser's set up to the 4th order. We deduced the expression to compute them directly from the normalised central moments η_{pq} , so they can easily be implemented with Summed-area tables. The complete set used in this work (11 moments) follows:

$$\psi_1 = \eta_{20} + \eta_{02} \quad (7)$$

$$\psi_2 = (\eta_{30} + \eta_{12})^2 + (\eta_{03} + \eta_{21})^2 \quad (8)$$

$$\psi_3 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12}) + [(\eta_{30} + \eta_{12})^2 - 3((\eta_{21} + \eta_{03})^2)] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \quad (9)$$

$$\psi_4 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - ((\eta_{21} + \eta_{03})^2) + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})] \quad (10)$$

$$\psi_5 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12}) [(\eta_{30} + \eta_{12})^2 - 3((\eta_{21} + \eta_{03})^2) + (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03}) - [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]] \quad (11)$$

$$\psi_6 = \eta_{11}((\eta_{30} + \eta_{12})^2 - (\eta_{03} + \eta_{21})^2) - (\eta_{20} - \eta_{02})(\eta_{30} + \eta_{12})(\eta_{03} + \eta_{21}) \quad (12)$$

$$\psi_7 = \eta_{40} + \eta_{04} + 2\eta_{22} \quad (13)$$

$$\psi_8 = (\eta_{40} - \eta_{04})[(\eta_{30} + \eta_{12})^2 - (\eta_{03} + \eta_{21})^2] + 4(\eta_{31} - \eta_{13})(\eta_{30} - \eta_{12})(\eta_{03} - \eta_{21}) \quad (14)$$

$$\psi_9 = 2(\eta_{31} + \eta_{13})[(\eta_{21} + \eta_{03})^2 - (\eta_{30} + \eta_{12})^2] + 2(\eta_{30} - \eta_{12})(\eta_{21} - \eta_{03})(\eta_{40} - \eta_{04}) \quad (15)$$

$$\psi_{10} = (\eta_{40} - 6\eta_{22} + \eta_{04}) \{[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]^2 - 4(\eta_{30} + \eta_{12})^2(\eta_{03} + \eta_{21})^2\} + 16(\eta_{31} - \eta_{13})(\eta_{30} + \eta_{12})(\eta_{03} + \eta_{21}) [(\eta_{30} + \eta_{12})^2 - (\eta_{03} + \eta_{21})^2] \quad (16)$$

$$\psi_{11} = 4(\eta_{40} - 6\eta_{22} + \eta_{04})(\eta_{30} + \eta_{12})(\eta_{03} + \eta_{21}) [(\eta_{30} + \eta_{12})^2 - (\eta_{03} + \eta_{21})^2] - 4(\eta_{31} - \eta_{13}) \{[(\eta_{30} + \eta_{12})^2 - (\eta_{03} + \eta_{21})^2]^2 - 4(\eta_{30} + \eta_{12})^2(\eta_{03} + \eta_{21})^2\} \quad (17)$$

These moments are invariant to translation, scaling, rotation and mirroring.

3.3 Computing Moment Invariants from Summed-area Tables

The similarity between equations 1 and 3 shows that one can compute 2-D moments directly from the Summed-area Tables. In order to create the tables we modify equation 2:

$$m_{pq}(x, y) = m_{pq}(x - 1, y) + m_{pq}I(x, y - 1) - m_{pq}(x - 1, y - 1) + i(x, y).x^p.y^q \quad (18)$$

One can find generically the 2-D moment $m_{p,q}(x', y', s)$ of a sub-window at (x', y') with scaling factor s . Lets consider two identical sub-windows that are located in different places in the image. The sub-windows would have different 2-D moments for orders $p \geq 0$ and $q \geq 0$. However, their values are equivalent to computing the moments based on the same sub-windows padded by pixels of value zero. As the Moment Invariants, ψ_n , are translation independent, their values are the same for both sub-windows.

The equations 7 to 17 depend only on the seven normalised central moments $\eta_{11}, \eta_{20}, \eta_{02}, \eta_{12}, \eta_{21}, \eta_{30}$ and η_{03} . These on the other hand depend on μ_{pq} which can be computed using a number of 2-D moments m_{pq} and therefore using Summed-area Tables directly. The zeroth order μ_{00} corresponds to the simplest Summed-area Table, i.e., the equivalent to the Integral Image used by Viola-Jones:

$$\mu_{00} = \sum_x \sum_y (\bar{x} - x)^0 (\bar{y} - y)^0 i(x, y) = m_{00} \quad (19)$$

For order 1,1 μ_{11} can be derived as follows:

$$\begin{aligned} \mu_{11} &= \sum_x \sum_y (x - \bar{x})^1 (y - \bar{y})^1 i(x, y) \\ &= \sum_x \sum_y (x.y - x.\bar{y} - \bar{x}.y + \bar{x}.\bar{y}).i(x, y) \\ &= \sum_x \sum_y x.y.i(x, y) - \sum_x \sum_y x.\bar{y}.i(x, y) \\ &\quad - \sum_x \sum_y \bar{x}.y.i(x, y) + \sum_x \sum_y \bar{x}.\bar{y}.i(x, y) \end{aligned} \quad (20)$$

Both \bar{x} and \bar{y} are constant for a sub-window. Also each of the four factors can be expressed as a function of the corresponding Summed-area Table:

$$\mu_{11} = m_{11} - \bar{y}.m_{10} - \bar{x}.m_{01} + \bar{x}.\bar{y}.m_{00} \quad (21)$$

We can now compute the central moment of order 1,1 for any sub-window based on the Summed-area Table computed over the entire image. Analogous derivation can be done to the other μ_{pq} , for which we only write the final equations as a function of m_{pq}, \bar{x} and \bar{y} here:

$$\mu_{20} = m_{20} - \bar{x}m_{10} \quad (22)$$

$$\mu_{02} = m_{02} - \bar{y}m_{01} \quad (23)$$

$$\mu_{30} = m_{30} - 3\bar{x}m_{20} + 2\bar{x}^2m_{10} \quad (24)$$

$$\mu_{03} = m_{03} - 3\bar{y}m_{02} + 2\bar{y}^2m_{01} \quad (25)$$

$$\mu_{12} = m_{12} - 2\bar{y}m_{11} - \bar{x}m_{02} + 2\bar{y}^2m_{10} \quad (26)$$

$$\mu_{21} = m_{21} - 2\bar{x}m_{11} - \bar{y}m_{20} + 2\bar{x}^2m_{01} \quad (27)$$

$$\mu_{22} = m_{22} - 2\bar{y}m_{21} + \bar{y}^2m_{20} - 2\bar{x}m_{12} + 4\bar{x}\bar{y}m_{11} - 2\bar{x}\bar{y}^2m_{10} + \bar{x}^2m_{02} - 2\bar{x}^2\bar{y}m_{01} + \bar{x}^2\bar{y}^2m_{00} \quad (28)$$

$$\mu_{31} = m_{31} - \bar{y}m_{30} + 3\bar{x}\bar{y}(m_{20} - m_{21}) + 3\bar{x}^2(m_{11} - \bar{y}m_{10}) + \bar{x}^3(\bar{y}m_{00} - m_{01}) \quad (29)$$

$$\mu_{13} = m_{13} - \bar{x}.m_{03} + 3\bar{x}\bar{y}(m_{02} - m_{12}) + 3\bar{y}^2(m_{11} - \bar{x}m_{01}) + \bar{y}^3(\bar{x}m_{00} - m_{10}) \quad (30)$$

$$\mu_{40} = m_{40} - 4\bar{x}m_{30} + 6\bar{x}^2m_{20} - 4\bar{x}^3m_{10} + \bar{x}^4m_{00} \quad (31)$$

$$\mu_{04} = m_{04} - 4\bar{y}m_{03} + 6\bar{y}^2m_{02} - 4\bar{y}^3m_{01} + \bar{y}^4m_{00} \quad (32)$$

The central moments needed for the 11 independent Moment Invariants ϕ_n can be computed from the following 15 Summed-area Tables: $m_{00}, m_{10}, m_{01}, m_{11}, m_{20}, m_{02}, m_{12}, m_{21}, m_{30}, m_{03}, m_{04}, m_{40}, m_{22}, m_{31}, m_{13}$.

4 Proposed method for feature extraction

Rotation and scaling invariance may be critical for the recognition task for some applications. In this section we describe how to increase the number of features by computing the moments of quasi-circular areas of the image. Using equations 7 to 17 we can compute the 11 moments from rectangular areas. Using an approximation we can compute the 11 moments from a circular area. For a particular area of interest we can proceed to compute circular areas defined by a series of concentric discs. This can be achieved using the same pre-computed 15 Summed-area tables, but it requires extra look-ups.

The resulting set of features will be invariant to rotation and scaling, but not translation. For applications that scan images to detect objects, such as in [3], this can be considered as an advantage. If the object being searched is over a dark background, many sub-window candidates will be found. However the concentric discs approach guaranties that only the few sub-windows that are centred on the object will match the pattern.

4.1 Circular Area (Discs)

Normally each rectangular sub-window requires only 4 table look-ups per Summed-area table (a total of 60 look-ups if we use the 15 Summed-area tables proposed here). In order to compute a circular sub-window, an approximation requires that small square areas are subtracted from the originally square sub-window (figure 1). The number of look-ups can be minimised because there are common points among the smaller square areas. Some of the points are not at all necessary because they are cancelled out. To compute the sum over a square area, the points 1,2,3 and 4 of a Summed-area Table are used as follows:

$$A_{square} = pt_1 - pt_2 - pt_3 + pt_4 \quad (33)$$

If the 12 squares (a,b, ... and l) are to be subtracted from the large square that defines the sub-window, we have A_{disc} :

$$\begin{aligned} A_{disc} = & pt_1 - pt_2 - pt_3 + pt_4 \\ & -(pt_{a1} - pt_{a2} - pt_{a3} + pt_{a4}) - (pt_{b1} - pt_{b2} - pt_{b3} + pt_{b4}) \\ & \dots - (pt_{k1} - pt_{k2} - pt_{k3} + pt_{k4}) - (pt_{l1} - pt_{l2} - pt_{l3} + pt_{l4}) \end{aligned} \quad (34)$$

But there are common points among the square areas that cancel each other. Rewriting the equation 34 as a function of duplicate points:

$$\begin{aligned} A_{disc} = & -pt_{a4} + pt_{b3} + pt_{c2} - pt_{d1} + pt_{e2} + pt_{e3} - pt_{e4} \\ & - pt_{f1} + pt_{f3} - pt_{f4} + pt_{g2} + pt_{g3} - pt_{g4} - pt_{h1} \\ & + pt_{h3} - pt_{h4} - pt_{i1} + pt_{i2} - pt_{i4} - pt_{j1} + pt_{j2} \\ & + pt_{j3} - pt_{k1} + pt_{k2} - pt_{k4} - pt_{l1} + pt_{l2} + pt_{l3} \end{aligned} \quad (35)$$

And therefore it suffices that 28 points are defined to compute the sum of pixels in figure 1 using all the 12 square areas (from a to l). Considering the 11 moments and that we need 15 Summed-area Tables, the total number of look-ups per sub-window for the 11 moments is 420.

In order to compare with similar strategies used in Haar-like feature extraction, a classifier produced with [3] method for face recognition implemented in OpenCV [12] has total of 2913 Haar-like features distributed in 24 layers (cascades), requiring 17478 look-ups. However not all sub-windows will reach the last layer, being eliminated by the classifier at a earlier stage. If a sub-window reaches the 9th layer it would have used around 500 look-ups, which is comparable to the method proposed in this work.

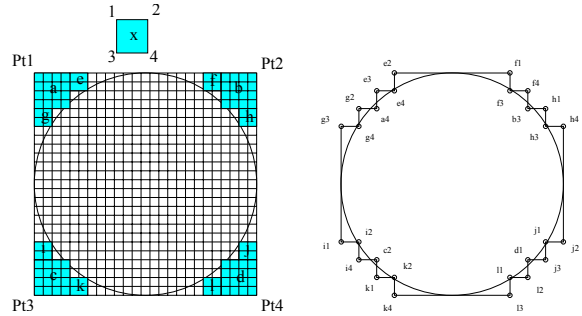


Figure 1: Computing the moments for an approximation of a circular area: the 12 square areas (in dark) are subtracted from each sub-window.

4.2 Concentric Discs

Finally, we can compute 11 moments for discs of different diameters. Figure 2 shows an example where a total of 66 moments are being computed. Each disc has its own pattern, as it catches different pixels of the image. The method has the potential to improve the discrimination powers of the feature set.



Figure 2: The concentric discs approach: more features are extracted, as the areas within the inner circles get different patterns, enriching the feature set.

The complete set of moments computed over concentric discs is obviously not translation invariant. If the centre of the concentric discs is moved the values for the internal discs will differ from the ones computed based on the previous centre. This can be used to locate the exact position of an object in the case of detection algorithms that use scanning and compare the values using classifiers (see for example the way sub-windows containing certain patterns can be located using Viola and Jones method [3]).

5 Experimental Results and Discussion



Figure 3: Test image from [10].

Experiments were carried out to verify the rotation and scaling invariance of the features extracted by this method. Firstly the image in figure 3 (which can be found in [10]) was used to compute the first 5 moments of Hu's set and the extra 6 moments from Flusser's set. The first 5 values might differ slightly from [10] due to the variable precision used to create the integral images and used to compute the set. For comparison, the logarithm of the absolute values of the moments were computed.

Table 1: 11 moments computed over a square area for image in figure 3.

	images					
	orig.	half	mir.	2°	45°	σ
ψ_1	6.600	6.600	6.600	6.596	6.595	0.0000
ψ_2	23.888	23.888	23.888	23.866	23.868	0.0001
ψ_3	49.200	49.201	49.200	49.152	49.134	0.0010
ψ_4	32.102	32.102	32.102	32.073	32.074	0.0002
ψ_5	47.850	47.850	47.850	47.807	47.810	0.0005
ψ_6	34.765	34.766	34.765	34.739	34.718	0.0005
ψ_7	12.838	12.838	12.838	12.830	12.829	0.0000
ψ_8	38.158	38.158	38.158	38.124	38.126	0.0003
ψ_9	40.248	40.250	40.248	40.220	40.197	0.0006
ψ_{10}	61.701	61.701	61.701	61.649	61.649	0.0008
ψ_{11}	61.978	61.978	61.978	61.930	61.924	0.0007

In a second experiment we used the same 5 images, but this time we computed the moments using the concentric discs approach. The discs' diameters are computed as a function of the width of the images

(e.g., here 0.9 means 90% of the width). Table 2 shows the results for the disc with a diameter of 50% of the width. Table 3 shows the variance of the results for discs of various diameters (as per figure 2).

Table 2: 11 moments computed over an approximation of circular area using 12 square areas (with a diameter of 50% of the width) for image in figure 3.

	images					
	orig.	half	mirr.	2°	45°	σ
ψ_1	6.615	6.615	6.615	6.611	6.613	0.0000
ψ_2	25.225	25.243	25.225	25.201	25.192	0.0004
ψ_3	50.889	50.948	50.889	50.857	51.702	0.1311
ψ_4	34.151	34.179	34.151	34.110	34.143	0.0006
ψ_5	50.352	50.390	50.352	50.293	50.454	0.0035
ψ_6	35.362	35.364	35.362	35.363	35.376	0.0001
ψ_7	12.943	12.942	12.943	12.934	12.938	0.0000
ψ_8	40.352	40.379	40.352	40.305	40.344	0.0007
ψ_9	40.910	40.909	40.910	40.918	40.945	0.0002
ψ_{10}	66.087	66.270	66.087	66.052	66.161	0.0077
ψ_{11}	67.417	67.407	67.417	67.811	67.425	0.0311

The results show that the approximation of the circular area is successful, as the variance is small for most moments. The set of features could be used for recognition tasks where the rotation invariance is important. The scaling invariance is maintained, as the approximation of the disc would be equivalent at different scales.

Figure 4 shows the actual approximations used to compute the discs for 50% diameter. The extra few areas outside the circle create some variation on the moments values. A better approximation could be easily implemented by subtracting more areas (if the accuracy is critical). However any improvement comes with the extra cost of more look-ups in the Summed-area tables.

Table 3: Variances for the concentric disc features from 1 to 0.5 in diameter.

	diameter					
	1	0.9	0.8	0.7	0.6	0.5
σ_{ψ_1}	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
σ_{ψ_2}	0.0001	0.0001	0.0002	0.0001	0.0001	0.0004
σ_{ψ_3}	0.0010	0.0003	0.0047	0.0015	0.0112	0.1311
σ_{ψ_4}	0.0002	0.0004	0.0012	0.0007	0.0047	0.0006
σ_{ψ_5}	0.0005	0.0007	0.0016	0.0026	0.0089	0.0035
σ_{ψ_6}	0.0005	0.0000	0.0047	0.4099	0.2052	0.0001
σ_{ψ_7}	0.0000	0.0000	0.0001	0.0000	0.0000	0.0000
σ_{ψ_8}	0.0003	0.0007	0.0021	0.0010	0.0078	0.0007
σ_{ψ_9}	0.0006	0.0007	0.0346	0.2224	1.6947	0.0002
$\sigma_{\psi_{10}}$	0.0008	0.0034	0.0221	0.2331	0.1935	0.0077
$\sigma_{\psi_{11}}$	0.0007	0.0014	0.0176	1.1691	0.0192	0.0311

As expected, the larger variances in table 3 are associated with the moments in higher dimensions. To compare the variances obtained with the approximation of a circular area, we cut circular areas (these are only as accurate as the scale permits) from the original images and measured the variance (see table 4). That would be the result if several smaller square areas were being subtracted from the image in such a way that the same pixels were involved in the computation of the moments.

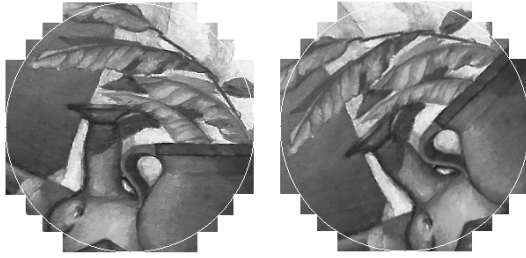


Figure 4: A 50% concentric disc area approximation for the original image and the 45° rotated image.

In other words, table 4 reflects the best case scenario for this set of images in the case of the concentric discs features.

Table 4: Variances for the concentric disc areas (cut directly from the images) from 1 to 0.5 in diameter.

	diameter					
	1	0.9	0.8	0.7	0.6	0.5
σ_{ψ_1}	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
σ_{ψ_2}	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
σ_{ψ_3}	0.0010	0.0007	0.0008	0.0007	0.0011	0.0004
σ_{ψ_4}	0.0002	0.0002	0.0003	0.0003	0.0003	0.0003
σ_{ψ_5}	0.0005	0.0006	0.0006	0.0006	0.0006	0.0007
σ_{ψ_6}	0.0005	0.0001	0.0002	0.0002	0.0003	0.0002
σ_{ψ_7}	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
σ_{ψ_8}	0.0003	0.0003	0.0004	0.0004	0.0004	0.0003
σ_{ψ_9}	0.0006	0.0001	0.0001	0.0005	0.0005	0.0003
$\sigma_{\psi_{10}}$	0.0008	0.0007	0.0009	0.0010	0.0008	0.0010
$\sigma_{\psi_{11}}$	0.0007	0.0007	0.0007	0.0005	0.0004	0.0011

It is beyond the scope of this paper to measure the speed of the method in details. As an indication, we have scanned an image to compute the 66 moments from sub-windows of the image 3. Using one processor of a dualcore AMD 2GHz we could compute close to 20000 complete moment sets per second (66 moments per sub-window at various scales and positions). However the code is not optimised and there is certainly room for improvement. The computation of a single moment is not faster than in other methods, but the computation of all the moments at different scales and positions benefits from the Summed-area table structure. It can be fast enough to allow real-time detection applications to use this method.

6 Conclusions and Future Work

We proposed a new method of feature extraction based on moments that maintains the rotation and the scaling invariance. Although sub-sets of the feature set are also translation invariant, the set as a whole is not invariant to translation and can be used to locate specific sub-windows via scanning.

An analysis of the accuracy taking moments from concentric discs was carried out. Most of the moments have a small variance when faced with rotation and scaling operations. Potentially the new

method is useful for object detection and recognition.

There are many unanswered questions regarding this new set of features, specially regarding the speed and accuracy in which classifiers can be trained using these features. We intend to further study the method by applying it to object detection and recognition tasks.

References

- [1] F. C. Crow, "Summed-area tables for texture mapping," *Computer Graphics*, vol. 18, pp. 207–212, July 1984.
- [2] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, pp. 179–187, 1962.
- [3] P. Viola and M. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, pp. 137–154, 2004.
- [4] J. Flusser, "On the independence of rotation moment invariants," *Pattern Recognition*, vol. 33, pp. 1405–1410, 2000.
- [5] H. Arof and F. Deravi, "Circular neighbourhood and 1-d dft features for texture classification and segmentation," *IEE Proceedings-Vision Image and Signal Processing*, vol. 145, pp. 167–172, Jun 1998.
- [6] L. Torres-Mendez, J. C. Ruiz-Suarez, L. E. Sucar, and G. Gomez, "Translation, rotation, and scale-invariant object recognition," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 30, pp. 125–130, February 2000.
- [7] M. Kazhdan, B. Chazelle, D. Dobkin, T. Funkhouser, and S. Rusinkiewicz, "A reflective symmetry descriptor for 3D models," *Algorithmica*, vol. 38, oct 2003.
- [8] R. Mukundan, "Radial tchebichef invariants for pattern recognition," in *Proc. of IEEE Tencon Conference Tencon05*, (Melbourne), pp. 2098–2103, Nov 2005.
- [9] C. Chong, P. Raveendran, and R. Mukundan, "A comparative analysis of algorithms for fast computation of zernike moments," *Pattern Recognition*, vol. 36, pp. 731–742, 2003.
- [10] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Prentice Hall, 2002.
- [11] A. K. Jain, *Fundamentals of Digital Image Processing*. Prentice Hall, 1989.
- [12] G. Bradski, "The opencv library," *Dr. Dobb's Journal*, pp. 120–126, November 2000.

A Robust Efficient Motion Segmentation Algorithm

Hongzhi Gao and Dr. Richard Green

Department of Computer Science and Software Engineering
University of Canterbury

{Hongzhi.Gao, Richard.Green}@canterbury.ac.nz

Abstract

The background subtraction double difference algorithm (BSDDA) is proposed in this paper. This algorithm is based on a combination of a background subtraction algorithm (BSA) and a double difference algorithm (DDA). BSDDA inherits advantages from both BSA and DDA. The results show that this approach is more robust to dynamic background scene than the BSA and it is more sensitive to slow moving objects than DDA.

Keywords: motion segmentation, background subtraction, moving object detection

1 Introduction

Background subtraction is used to segment motion in a static scene. Its name indicates the most intuitive way of achieving this goal: subtracting background from the observed image and the residue of this process (motion residue) is the segmented moving objects. A significant amount of recent research [1, 2] attempts to develop good models for segmenting background scenes accurately in computer vision (CV) systems.

As Cucchiara *et. al.* pointed out in [6], a good background model should solve two problems. The first problem is that the model should reflect the real background as accurately as possible. Secondly, the model should immediately update when the scene changes. It is not easy to determine the optimal solution that will satisfy both problems at the same time. The best trade offs for both problems significantly rely on the use case of CV systems customised to particular data sets.

For example, [14] introduces a handball player tracking application. In this system, cameras are securely mounted and lighting conditions are fixed. In this constrained environment, the original background subtraction algorithm works well.

A football tracking system is discussed in [15]. This application handles broadcasted digital television signal that contains a lot of panning, zooming and scene switching. The author selects adjacent frame difference algorithm to detect moving balls so that the motion residue introduced by background scene variation could be minimized.

In video surveillance systems, such as [3, 4 and 15], long term adaptation of the slow background variation is a critical requirement. In these projects, statistic background models (SBM) such as a Gaussian background model, are applied. Such background models describe each pixel in the background scene as a set of values following Gaussian probably distribution function. By updating factors such as mean and variation, of the Gaussian function with new observation, the problem of slow scene variation caused by time of the day or weather condition can be solved.

In addition to the Gaussian background models (GBM), multi-valued background models have been introduced to handle waving tree branches. Some recent projects [8, 9 and 12] use mixture of multiple Gaussian distributions to model the background scenes. Compared with GBM, these improved models give a better representation of periodical background scene variation.

The idea of our improved motion segmentation algorithm (MSA) came from CV research [15, 16] that tracks moving objects in outdoor environments. This system has the following constraints: firstly, as it is a real time CV system (i.e. 30+fps), the algorithm must be simple; secondly, it must handle rapid changes (usually caused by camera shaking) and slow changes (usually caused by weather or time of the day) and so various background scene variations need to be allowed for; and finally, the algorithm used in this application should work well with relatively large slow moving objects, such as a human body. In the

rest of this paper, we will show how our improved algorithm satisfies these requirements.

This paper is organized as follows. In the next section, some related works will be briefly reviewed. In section three, we present how our algorithm is made. In section four, we present experimental results which show that this improved algorithm satisfies the above requirements. Finally, the last section discusses future research directions and concludes this paper.

2 Related Work

Many motion segmentation research projects in the past focused on constructing and maintaining a good background model. Considering the information source, these research works can be categorized into three classes. The idea of this categorization is illustrated in figure 1.

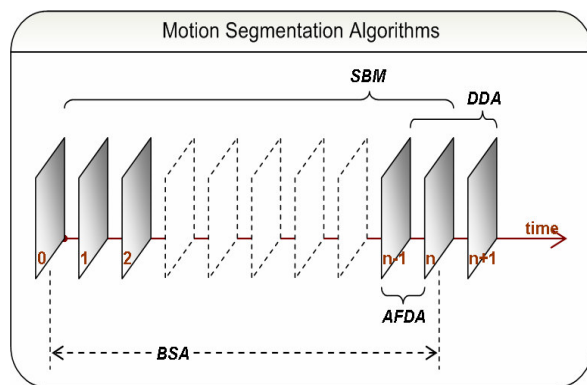


Figure 1: A coarse classification of the background model research works in the literature.

In figure 1, the horizontal axis indicates time. It begins at left side, which indicates the starting time of video capture, and extends toward right side, which indicates the latest captured image so far. Each frame of the image in this figure has a unique sequence number, which is marked at the corner of the image frames in Figure 1. This sequence number starts from zero and ends at 'n+1' which indicates the latest captured image.

The most intuitive and simple background model uses the first captured image (the frame with sequence number zero) to describe the background scene. This background image is then subtracted from each subsequent image captured by the camera. Moving objects can then be identified by the difference of the two images. The most significant drawback of this algorithm is that the environment must be strictly controlled so that the background scene remains the same as the first image that was captured and used as the reference frame.

To be used in unconstrained environment, [14 and 17] model background scene as the image that is immediately preceding the current one. For example,

if the frame with sequence number 'n' in figure 1 is the 'current' image, in [14 and 17], the background model is given by frame 'n-1'. The differences between these two images indicate the place where movement is happened. This approach is called adjacent frame difference algorithm. Comparing with BSA, this algorithm is more robust to the dynamic scene; however it introduces the 'ghost' objects problem discussed in [18].

Double difference algorithm (DDA) is an improved adjacent frame difference approach. It defines the second last image in a sequence (frame n in figure 1) as the 'current' frame. Instead of using one adjacent frame (n-1), it lets adjacent frames at both side (frame n-1 and frame n+1) become background images. In this algorithm, two difference images, d1 and d2 are calculated by subtracting the 'current' frame n from both background frames (n-1 and n+1) separately. The common residue area in the two difference images (d1 and d2) are the detected moving objects. DDA solves the 'ghost' object [18] problem, however, research works in [15 and 16] show that DDA is not suitable to detect large and slow moving objects, for example human activity, as it outputs many disconnected clusters along the object's contour (figure 2).



Figure 2: This figure shows a result generated with DDA. As shown, a birds-eye view of the human silhouette holding a bat is hollow and not well segmented.

Many research works [3, 4, 5, 6, 7, 9, 11 and 12] have been done in the statistic background model (SBM) domain to find answers to the background subtraction problem. This research can be classified together in one category as they use information of the whole past images (figure 1) to construct a background representation.

Gaussian background model (GBM) was proposed by Wren *et.al.* in [3]. Each pixel in the background image is modeled independently by fitting the past values of pixels at the same location in a Gaussian probability density function. When a new frame comes, the new pixel value will be judged by confidence interval [19] with the existing GBM to distinguish whether it belongs to the background or

foreground object. Although GBM is very popular [20], it lacks ability to deal with multi-valued backgrounds caused by dynamic nature of real world scenes [12 and 20]. For example, a natural environment may include swaying vegetation, rippling water, and flickering monitors and such cases are unable to be modelled as a single Gaussian distribution.

Mixture of multiple Gaussian background model is proposed by [12] to model the dynamic nature of real world scene. In this research, the value of each pixel in the background is modelled as set of (usually 3 to 5) separate Gaussian distributions. Each distribution represents one possible value of this pixel. The pixel value in the new image frame is compared against all these distributions [8] to distinguish between background and foreground. However, as argued in [5, 11, 13 and 16], this model is complex and less efficient.

Apart from Gaussian based approaches, other SBM algorithms, for example median value [5, 6 and 11] and min-max values [7], have also been developed. However, most of these approaches suffer from a slow background model updating rate. Sudden scene variations, which may be caused by a camera shaking or simply turning on or off a light switch, will cause these algorithms to fail.

As introduced in the first section, an MSA is needed that is of low complexity and robust to both camera shaking and slow scene variations at the same time. It also needs to be sensitive to slow movements.

3 BSDDA

In our research, the MSA should satisfy the following requirements:

1. Sensitive to slow movements.
2. Output a complete silhouette for large slow moving objects, such as human body.
3. Robust in both slow (time of the day) and fast (camera shaking) background variation.
4. Simple and efficient.

We did an evaluation with the existing motion segmentation approaches in the literature. Although, BSA is simple, sensitive to both fast and slow motion, and able to output a complete silhouette for moving objects, this algorithm requires a very stable background scene, which doesn't satisfy our third requirement. On the other hand, although DDA is robust in dynamic background scene, it is not sensitive to slow moving objects and the detected results for large objects are usually incomplete (Figure 2). Finally, apart from the complexity, SBM based algorithms lack the ability to deal with sudden

background variations that may be caused by camera shaking.

To satisfy the four requirements listed above, a new motion segmentation technique is made from the combination of both BSA and DDA. Our approach has advantages that inherit from both algorithms and therefore, it is named the background subtraction double difference algorithm (BSDDA).

Figure 3 illustrates the four main processing steps of our approach.

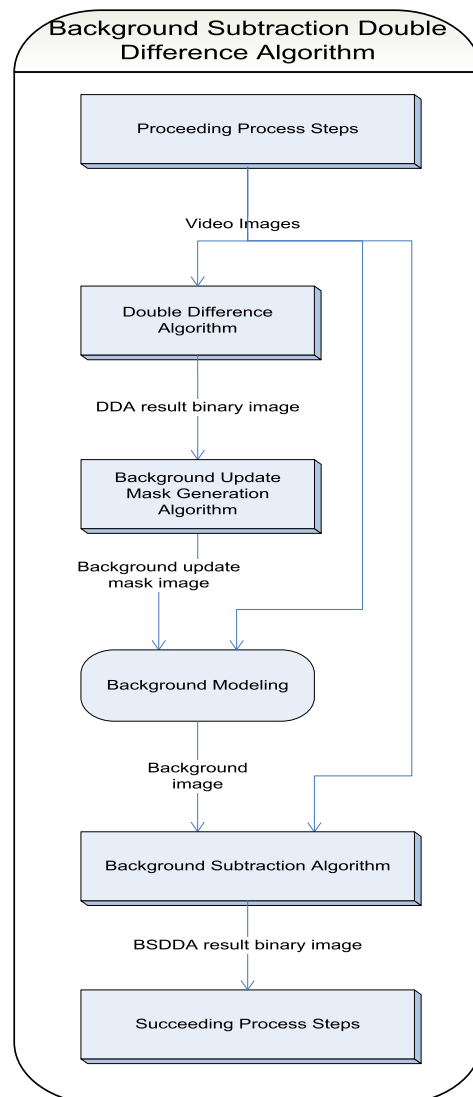


Figure 3: The main workflow of the background subtraction double difference algorithm.

Firstly, an image (Figure 4, top-left) is sent to DDA module by proceeding process modules in a CV system. The DDA module use ordinary DDA to generate a motion detection result; we name this result image as DDAR (Figure 2).

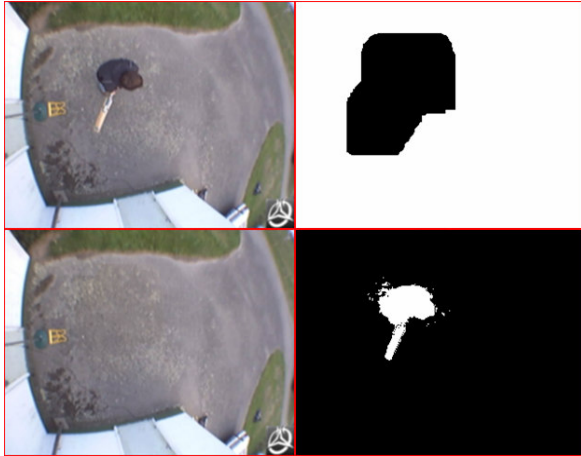


Figure 4: Working process of the BSDDA. Top-left: the original video input; top-right: the background update mask; bottom-left: the background model image; bottom-right: the final result of BSDDA.

Secondly, DDAR is sent to background update mask generate module. In this module, DDAR is converted to background update mask (Figure 4, top-right) corresponding to the current image. This mask is a binary image that has values of zero and one. The zero value means that foreground objects may be detected at this location in the original image and the one value means that this location in the original image is the background.

Thirdly, both the new image and the background update mask are sent to background modelling module. In this module, a pixel in the background model image (Figure 4, bottom-left) is updated with the value in the new image if its corresponding pixel location in BUM is one.

Finally, both the new image and the updated background model are sent to BSA module. In this BSA module, background subtraction is carried with two images. Its output is sent to successive modules as the final result image of BSDDA (Figure 4, bottom-right).

Figure 5 illustrates the algorithm that is used to generate BUM from DDAR.

As shown, the first step is to dilate DDAR. The shape and dimension of the dilation core used in this process are empirical values that are chosen based on image size and application. In our research, a 50 by 50 square is used. The dilation step works like a magnifier in BSDDA. It zooms in the output from DDA because DDA is less sensitive to slow moving large objects.

After dilation, the dilated DDAR (a binary image) is inverted so that the 'one' values indicate background pixels without motion and 'zero' values indicate detected moving objects (Figure 4, top-right). This

inversion step is to simplify the processes in the background update model.

As introduced above, unlike SBM based algorithms that use a statistical accumulation of the past background variations, our approach, BSDDA, is directly made from the combination of DDA and the BSA. DDA outputs a mask image for selective background model updating and the ordinary BSA is used to generate the final motion segmentation result.

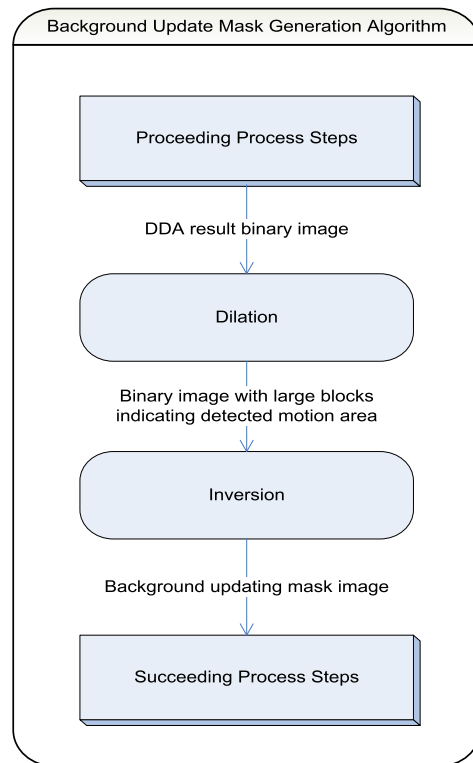


Figure 5: The workflow of generating background updating mask in BSDDA.

4 Experimental Result

To evaluate BSDDA, it has been tested using outdoor scene video clips.

The test video clips are taken by a Logitech QuickCam 4000 camera with a wide angle (130 degrees) lens. The camera, facing downward, is mounted on top of a mast of 3.5 meters height. The content of this video is a cricket game. The batter stands directly under the camera and the bowler is out of camera's view.

Figure 6 shows a screen-snap of the testing system. There are four images illustrated in this figure: the top-left one is original image captured from the camera; the top-right one is the motion residue detected by BSA; the bottom-left one is the DDA motion detection result; the bottom-right one is the motion segmentation output of BSDDA.

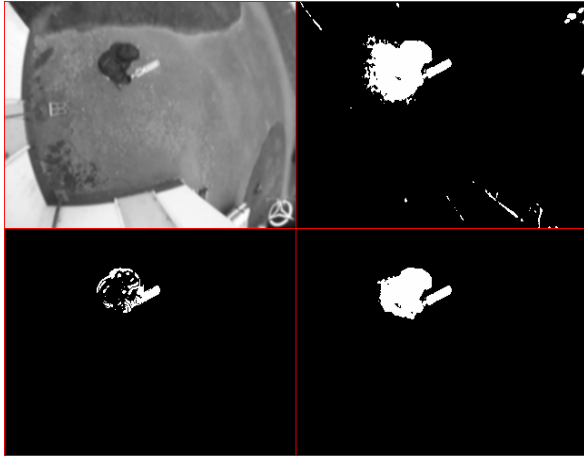


Figure 6: A screen snap of the BSDDA testing system.

A pixel counting experiment was designed to calculate the number of segmented pixels in the segmentation results from the BSA, DDA and BSDDA in order to compare the quality of these algorithms.

In this experiment, the motion residue in the segmented masks is divided into two categories manually. If a pixel in the mask image indicates moving objects in the test footage, this pixel is put into the “Motion” category; on the other hand, if the pixel is triggered by noise, it would be put into the “Noise” category. The numbers of pixels in both categories are counted and the results are shown in Table 1.

Figure 7 illustrates the number of moving pixels detected by BSA, DDA and BSDDA. In this diagram, the X axis is the frame number and the Y axis is the number of segmented pixels in the “Motion” category. The curve represented by round dots shows the motion segmentation result of BSA. The motion segment result of BSA gives the most complete silhouette of the moving object. The BSDDA segmentation result (shown in the curve with triangular dots) is consistently adhering to the BSA result in terms of silhouette completeness.

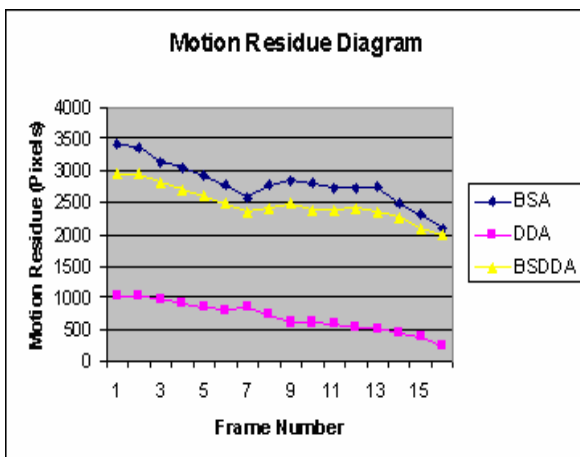


Figure 7: Motion residue diagram

The number of noise pixels detected by BSA, DDA and BSDDA is shown in Figure 8. In this diagram, the X axis is the frame number and the Y axis is the number of segmented pixels triggered by noise. The noise pixels detected by BSDDA (represented by the curve with triangular dots) is consistently adhere to the DDA result, which is most robust to dynamic background variations. The peak in this diagram indicates a sudden background scene variation which is caused by camera shaking.

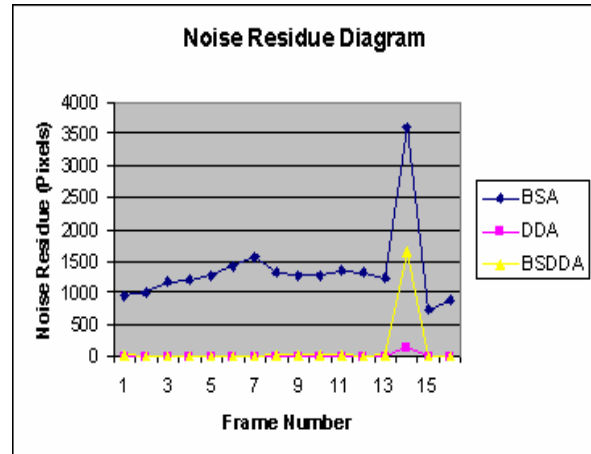


Figure 8: Noise residue diagram

5 Conclusion and Future Work

In this paper, we present the BSDDA algorithm which utilizes a combination of BSA and DDA to output complete silhouettes for slow moving large objects (i.e. human movement) and is robust to dynamic background scene at the same time. Compared with SBM based algorithms, our proposed approach is much less complex.

In the future, this research could be improved in two directions.

Firstly, in our approach once the foreground object stops moving but stays in the scene, it will become part of the background model immediately which may not always be the preferred outcome. SBM based algorithms solve this problem by introducing a learning factor to update the background model accumulatively. However, more research is needed to adaptively update various regions in a background model selectively.

Secondly, as discussed in the paper, the dilation step amplifies DDA results. However, this process magnifies both the motion residue and noises at the same time and so further research is needed to selectively magnify motion residue only.

Frame	BSA			DDA			BSDDA		
	Motion	Noise	N2M Ratio	Motion	Noise	N2M Ratio	Motion	Noise	N2M Ratio
222	3404	965	0.2835	1040	0	0	2944	20	0.0068
223	3365	1011	0.3004	1033	0	0	2936	12	0.0041
224	3145	1171	0.3723	979	0	0	2833	4	0.0014
225	3031	1200	0.3959	913	0	0	2693	4	0.0015
226	2914	1278	0.4386	857	0	0	2605	6	0.0023
227	2774	1435	0.5173	815	0	0	2485	2	0.0008
228	2572	1578	0.6135	853	0	0	2353	8	0.0034
229	2773	1333	0.4807	728	0	0	2406	13	0.0054
230	2838	1270	0.4475	606	0	0	2467	18	0.0073
231	2796	1287	0.4603	622	0	0	2387	14	0.0059
232	2715	1350	0.4972	592	0	0	2378	15	0.0063
233	2725	1322	0.4851	551	0	0	2398	9	0.0038
234	2750	1234	0.4487	517	0	0	2362	16	0.0068
235	2472	3618	1.4636	449	154	0.3430	2257	1635	0.7244
236	2310	733	0.3173	383	0	0	2083	0	0.0000
237	2085	881	0.4225	256	0	0	1979	8	0.0040

Table 1: Experiment results of BSA, DDA and BSDDA motion segmentation.

6 References

- [1] McIvor, A. M.; Background Subtraction Techniques; Proc. of Image and Vision Computing New Zealand; page 23-28; 2000.
- [2] Piccardi, M.; Background Subtraction Techniques: a Review; IEEE Int. Conf. on System, Man and Cybernetics; pages. 3099-3104; 2004.
- [3] Wren, C.; Azarhayejani, A.; Darrell, A.; Pentland, A. P.; Pfinder: real-time tracking of the human body; IEEE Trans. On Pattern Analysis and Machine Intelligence; pages: 780-785; 1997.
- [4] Koller, D.; Weber J.; Huang T.; Malik J.; Ogasawara, G.; Rao, B.; Russell, S.; Towards Robust Automatic Traffic Scene Analysis in Real-time; IEEE Int. Conf. on Pattern Recognition; pages: 126-131; 1994.
- [5] Lo, B. P. L.; Velastin, S. A.; Automatic Congestion Detection System for Underground Platforms; IEEE Int. Sym. on Intelligent Multimedia, Video and Speech Processing; pages: 158-161; 2001.
- [6] Cucchiara, R.; Grana, C.; Piccardi, M.; Prati, A.; Detecting Moving Objects, Ghosts and Shadows in Video Stream; IEEE Trans. on Pattern Analysis and Machine Intelligence; page 1337-1442; 2003.
- [7] Haritaoglu, I.; Harwood, D.; Davis, L. S.; W4: Real-Time Surveillance of People and Their Activities; IEEE Tran. on Pattern Analysis and Machine Intelligence; page 809-830; 2000.
- [8] Stauffer, C.; Grimson, W. E. L.; Learning Patterns of Activity Using Real-Time Tracking; IEEE Tran. on Pattern Analysis and Machine Intelligence; page 747-757; 2000.
- [9] Mckenna, S. J.; Jabri, S.; Duric, Z.; Rosenfeld, A.; Wechsler, H.; Tracking Groups of People; Conf. on Computer Vision and Image Understanding; page 42-56; 2000.
- [10] Ohta, N.; A Statistical Approach to Background Suppression for Surveillance System; IEEE Int. Conf. on Computer Vision; page 481-486; 2001.
- [11] Gloyer, B.; Aghajan, H. K.; Siu, K. Y.; Kailath, T.; Video-Based Freeway Monitoring System Using Recursive Vehicle Tracking; Proc. SPIE Symp. Electronic Imaging: Image and Video Processing; 1995.
- [12] Stauffer, C.; Grimson, W. E. L.; Adaptive Background Mixture Models for Real-Time Tracking; IEEE Int. Conf. on Computer Vision and Pattern Recognition; page 246-252; 1999.
- [13] Zhang, Q.; Klette, R.; Robust Background Subtraction and Maintenance; IEEE Conf. on Pattern Recognition; page 90-93; 2004.
- [14] Wedge, D.; Huynh, D.; Koves, P.; Tracking Footballs Through Clutter in Broadcast Digital Videos; In Proc. of Image and Vision Computing New Zealand; page 155-160; 2004.
- [15] Gao, H.; Green, R.; State-based Ball Detection and Tracking for Cricket Training; Proc. of Image and Vision Computing New Zealand; page 423-428; 2004.
- [16] Gao, H.; Tracking small, fast objects in noisy images; M.Sc. Thesis, University of Canterbury; 2005.
- [17] Lipton, A. J.; Fujiyoshi, H.; Patil, R. S.; Moving Target Classification and Tracking from Real-time Video; IEEE Workshop on Applications of Computer Vision; page 8-14; 1998.
- [18] Milan, S.; Vaclav, H.; Roger, B.; Image Processing, Analysis, and Machine Vision; Brooks and Cole Publishing; page 377-378; 1998.
- [19] Wolfram MathWorld; Confidence Interval; <http://mathworld.wolfram.com/ConfidenceInterval.html>; September. 2006.
- [20] Heikkila, M.; Pietikainen, M.; A Texture-Based Method for Modeling the Background and Detecting Moving Objects; IEEE Tran. on Pattern Analysis and Machine Intelligence; page 657-662; 2006.

Camera Egomotion Tracking using Markers

Brendon Kelly, Richard Green

University of Canterbury, Department of Computer Science and Software Engineering
bsk14@student.canterbury.ac.nz
richard.green@canterbury.ac.nz

Abstract

This paper investigates the performance of camera egomotion tracking using fixed markers. Tracking is performed using two different marker-based tracking systems. We also investigate the feasibility of using multiple markers for camera egomotion tracking, and propose a novel algorithm which can be used to devise the most efficient marker placement strategy for use with a multiple-marker based camera egomotion tracking system. Our results show that markers are useful for camera egomotion tracking within certain visual angle constraints.

Keywords: egomotion, augmented reality, ARToolKit, ARToolKitPlus

1 Introduction

Camera egomotion tracking is an important part of many computer vision fields, including Augmented Reality and vehicle guidance. Various methods of camera tracking have been proposed, ranging from purely optical techniques[1][2][3][4], such as optical flow tracking, to hybrid techniques[5][6], utilising GPS, ultrasonic and magnetic techniques as well as optical.

In this paper we investigate the use of a marker based tracking system, ARToolKit[7] for camera position tracking. The ARToolKit system is generally used to find the 6DOF position of markers relative to a camera, and therefore if these markers are fixed, we can use ARToolKit ‘in reverse’, by inverting the matrix used to represent each marker. This will give us a 6DOF position for the camera, with respect to a marker, whose position we already know. This inversion of the matrix of course, can cause high levels of tracking error at greater distances. We also use a second augmented reality system, ARToolKitPlus[8], which was, in part, developed to improve the tracking abilities of ARToolKit. It uses similar markers to ARToolKit, but markers require no training, as the marker identifier is embedded in the pattern itself.

Further to this, we investigate the efficient use of multiple markers at once to calculate camera position. The hypothesis here is that the greater the number of markers, the greater the reduction in error that may be induced by the large distances we intend to track over.

Finally, we introduce a novel algorithm to determine appropriate placement of ARToolKit style

markers for camera tracking. This is important, as a layout which is too sparse will mean the camera can move to positions where no markers are visible, and a layout which is too dense will be impractical, and cause significant visual pollution of the work space.

2 Background

ARToolKit is a software library that can be used to calculate camera position and orientation relative to physical markers in real time. This enables the easy development of a wide range of Augmented Reality applications. ARToolKitPlus is an extended version of ARToolKit’s vision code that adds new features, but breaks compatibility due to its class-based API. The extensions made in ARToolKitPlus include implementation of the “Robust Planar Pose” (RPP) algorithm. The RPP algorithm is used to give a more stable tracking than ARToolKit’s pose estimation algorithm.

Previous research into the range of ARToolKit markers[9][10] have been restricted to a distance of 3 metres, while we intend to investigate distances of up to 6 metres. Results from this research showed positioning errors of up to 20%.

The “living-room”[11] experiment involved the use of large (approximately 40cm by 40cm) markers printed on strips, which were then used to entirely wallpaper a 3m by 3m room. The room was used for exploring interactive, space-related aspects of augmented-reality. The researchers found that ARToolKit had severe shortcomings regarding precision and steadiness when used for camera tracking.



Figure 1: Two ARToolkit markers being used to calculate camera position. Markers are equal distances from ceiling and floor, and also the walls and each other.

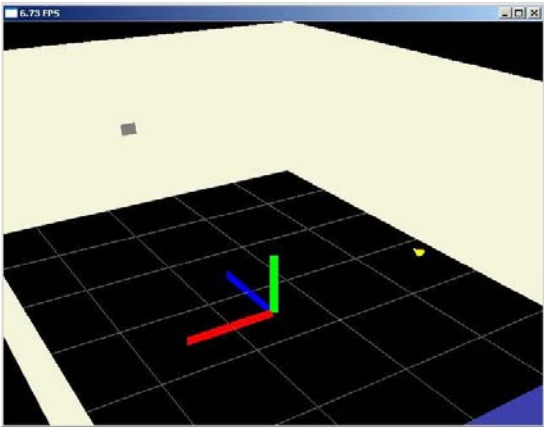


Figure 2: OpenGL representation of the room, including wall, marker and camera position, which is indicated by the yellow cone.

3 Method

The experiments conducted to evaluate the performance of our tracking systems consist of a series of accuracy and stability tests conducted using a webcam moving about a room. The room measures 6.6m by 5.5m by 3.0m and markers of size 20cm by 20cm are positioned at various points on the walls. Although bigger markers will provide greater accuracy and range for our system, we have constrained the marker size to be within the bounds of a standard A4 piece of paper, to ensure ease of marker creation. A single, standard webcam is used, running at a resolution of 640x480 pixels. An OpenGL model of the vision laboratory was used to represent the wall, marker and calculated camera position.

3.1 Multiple Markers

Our multiple marker experiments used the AR-Toolkit system, with 2 markers positioned on a 5.50m long by 3.0m high wall. The markers were both positioned at 1.5 m high, each 1/3 (1.83m) of the way from the end of the walls. The camera was positioned at a height of 1.25m, directly in line with the a point bisecting the two markers. The application developed for these experiments provided position values in millimetres, relative to the very centre of the room, at floor level. This meant that our first position was at $X = 0$, $Y = 1250$, $Z = 1000$. This was the closest we could get the camera, while still being able to identify both markers. In each subsequent recording we decreased the Z distance by 0.5 metres, until marker tracking was no longer achieved. At each position, 1000 calculated position values were taken. To analyse results, we calculated the mean value returned to measure accuracy, and the standard deviation in returned values to measure jitter.

3.2 ARToolkit vs ARToolkitPlus

These experiments were performed with a single marker placed at the central position of a 5.5m by 3m wall. The camera was positioned at a height of 1.25m, directly in line with the marker. The application developed for these experiments provided position values in millimetres, relative to the very centre of the room, at floor level. This meant that our first position was at $X = 0$, $Y = 1250$, $Z = 2000$. In each subsequent recording we decreased the Z distance by 0.5 metres, until marker tracking was no longer achieved. At each position, 1000 calculated position values were taken. To analyse results, we calculated the mean value returned to measure accuracy, and the standard deviation in returned values to measure jitter. The only change required between ARToolkit and ARToolkitPlus experiments was the use of a different marker, which was positioned in exactly the same place. Two settings were used when evaluating ARToolkitPlus; one using the standard ARToolkit pose estimator, and one using the RPP algorithm included with ARToolkitPlus. Thresholding of the camera image is an important part of marker detection in both ARToolkit and ARToolkitPlus. When conducting ARToolkitPlus experiments, its automatic thresholding feature was used. As ARToolkit does not have an automatic thresholding feature, this threshold was manually set.

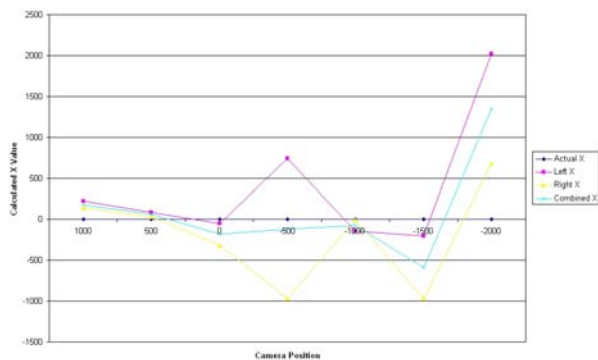


Figure 3: Results comparing 2 marker setup accuracy in the X dimension.

4 Evaluation

Initial investigation into the performance of ARToolKit showed a drop in tracking performance once distance from the marker exceeded 2m. Tracking became significantly less accurate, with a large amount of jitter. This jitter was also dependent on the camera's relative position with regard to the marker, so at some positions the estimation was stable, at others it was highly unstable. This was clearly evident when using more than one marker for position estimation. When using two markers, one would be far more stable than the other, but this relationship could be inverted by only a small change in camera position. This jitter was most evident in the X and Y values, while the Z values (distance from marker), showed significantly less jitter. These Z values were also far more accurate at range than values in the other two dimensions.

It is important to note that the accuracy and jitter levels for position calculation do not share a linear relationship with marker range. Once the size of the marker in the camera image, sometimes known as the visual angle, drops below a certain threshold, certain positions will yield accurate tracking and low jitter, while a position slightly closer to the marker may yield far poorer results. It is suggested that this is due to the relatively low resolution of the camera being used, and that the camera image of the marker at range may be significantly distorted by pixelisation. This fact in itself may mean that any long range marker detection with marker of this size is not practical without an increase in camera resolution.

In comparing our three pose estimators, we gained similar results to our 2 marker setup. All estimators suffered from significant jitter, except in the Z dimension, where accuracy was also significantly better. The RPP algorithm showed no significant increase in accuracy or decrease in jitter.

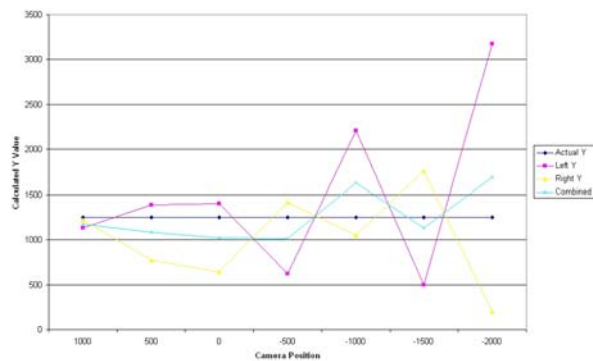


Figure 4: Results comparing 2 marker setup accuracy in the Y dimension.

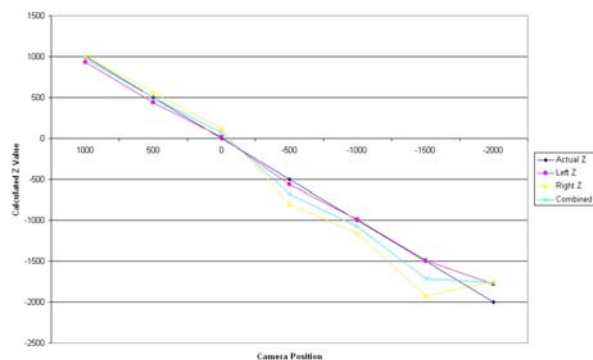


Figure 5: Results comparing 2 marker setup accuracy in the Z dimension.

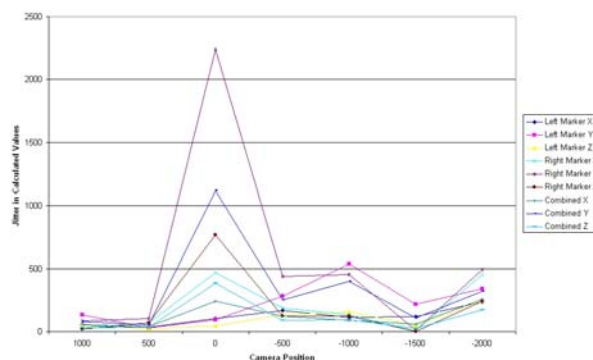


Figure 6: Results comparing 2 marker jitter levels in all dimensions.

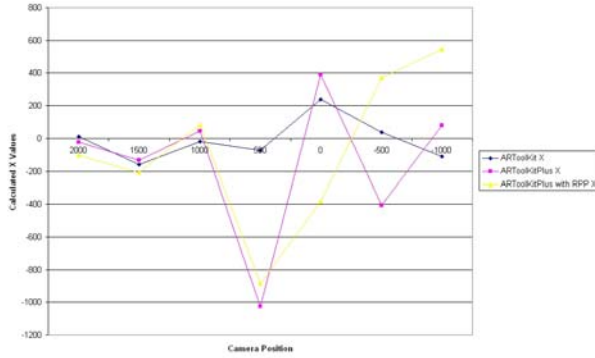


Figure 7: Results comparing all 3 pose estimator's accuracy in the X dimension.

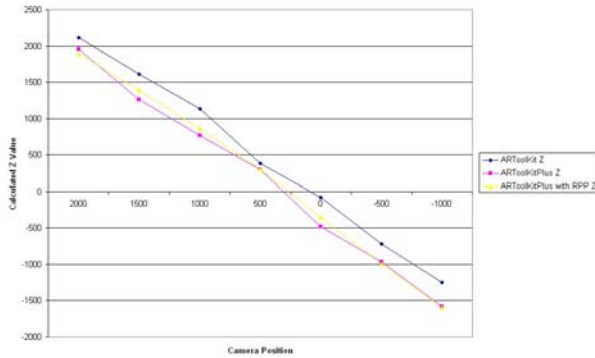


Figure 8: Results comparing all 3 pose estimator's accuracy in the Z dimension.

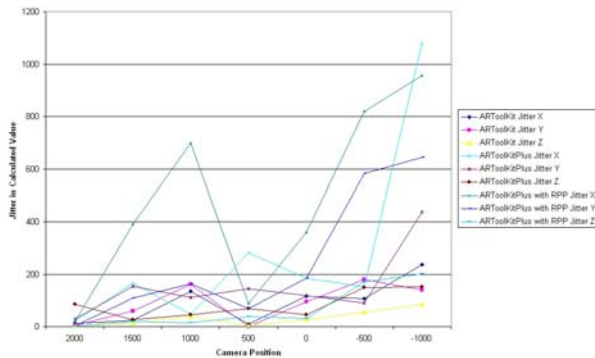


Figure 9: Results comparing all 3 pose estimator's jitter levels in all dimensions.

It was also noted that this algorithm was significantly slower than the standard ARToolKit pose estimator.

We found ARToolKit to be more effective when detecting markers and conducting pose estimation, but the difference is only significant when reaching the limits of marker detection for both systems. We considered marker tracking accuracy and jitter levels to be acceptable for both ARToolKit and

ARToolKitPlus, up to $Z = 500$ or $Z = 0$. These values represent a distance from marker of 2.8m and 3.3m respectively, and we propose that a distance of 3.0m represents the maximum range for markers of this size¹.

5 Marker Placement

To complement the marker tracking system evaluation, we propose an algorithm for placement of markers in a room environment. By taking into account predefined limits on camera movement, we can define the appropriate marker size and spacing to ensure reliable tracking at all points.

5.1 Minimum Marker Size

From our evaluation we have estimated an effective range of use for a 200mm by 200mm marker to be approximately 3.0m. Using these values we can calculate a minimum marker size ($minMS$) based on a required maximum distance ($maxD$), which would in most cases be the length of the longest wall in the room:

$$minMS = \frac{maxD}{15} \quad (1)$$

This value $minMS$ should be used when calculating the length of the sides of required markers.

5.2 Marker Separation

Once we have established a marker size (MS), we can calculate the maximum horizontal and vertical separations ($maxHS$ & $maxVS$), based on the camera's horizontal and vertical fields of view ($HFOV$ & $VFOV$), and a required minimum distance ($minD$), which can be defined as the minimum distance to any wall that the camera will move to:

$$maxHS = 2 \times \left(\left(minD \times \tan \left(\frac{HFOV}{2} \right) \right) - MS \right) \quad (2)$$

To calculate $maxVS$, $HFOV$ is replaced with $VFOV$.

These values should be used when setting the distances between edges of markers in the horizontal and vertical directions.

¹This is based on a webcam running 640 by 480 pixels resolution. This range may be different for cameras of a different resolution. See 'Further Work'.

6 Conclusion

We have shown that ARToolKit style markers may be useful for camera egomotion tracking, but in a practical situation, the tracking markers must be within ($15 \times \text{markersize}$) of the camera. Beyond this distance, calculation of X and Y coordinates become very unstable and innaccurate. Conversely, calculations for Z coordinates (the distance from marker) maintain accuracy and stability all the way out to the marker's maximum detectable position.

Multiple markers can provide greater tracking accuracy, but not when combined. In any practical application, the best approach to using multiple markers with this amount of variable jitter, is to find the marker with the highest 'confidence level', and use this marker for tracking in the current frame. This 'best confidence' technique is the most feasible approach.

ARToolKitPlus provides no significant improvements to ARToolKit when utilised for the purpose of camera egomotion tracking. In fact, ARToolKitPlus suffers from a more limited range, and subsequent reduction in accuracy and increase in jitter. This is probably caused by the different marker style, which does not include a solid white square segment, as ARToolKit markers do. This means the markers are more difficult to identify from their surroundings, and makes pose estimation less accurate.

We have proposed a marker placement algorithm, which can be used to devise the most efficient marker placement strategy for use with an ARToolKit style marker based camera egomotion tracking system. By using this algorithm, an efficient set of marker positions can be created, once the necessary marker size has been calculated for the room being used.

7 Further Work

As mentioned previously, thresholding of the image is an important part of marker detection. It was noted with both detection systems that adjustment of this threshold could markedly increase/decrease the performance of marker detection and pose estimation. It would be beneficial to investigate an automatic thresholding algorithm specifically designed for long-range marker tracking, as this may well extend detection and pose estimation range beyond the 3.0m limit we have proposed.

We also discussed the effect of marker size and camera resolution on marker detection range. An investigation into the relationship between marker size, camera resolution and marker detection range

could prove very interesting, as it is likely an increase in camera resolution may be equivalent to an increase in marker size, and could significantly increase this range. For instance, if camera resolution was doubled, this could double the effective range of a marker, making modestly sized markers (perhaps A4 paper sized), viable for tracking in a large room. In this paper we only evaluated position calculations, with no regard to orientation. Any more in-depth work would also benefit from an evaluation of this component.

As part of this paper, we proposed a marker placement algorithm, based on results obtained during our research. A thorough evaluation of this algorithm has not been done, and it is also left open for extension, including the possible need for extended camera parameters. As discussed previously an improved camera resolution may mean smaller markers are feasible, resulting in less visual pollution of the workspace.

References

- [1] A.-T. Tsao, C.-S. Fuh, Y.-P. Hung, and Y.-S. Chen, "Ego-motion estimation using optical flow fields observed from multiple cameras," in *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, (Washington, DC, USA), p. 457, IEEE Computer Society, 1997.
- [2] D. Koller, G. Klinker, E. Rose, D. Breen, R. Whitaker, and M. Tuceryan, "Real-time Vision-Based camera tracking for augmented reality applications," in *ACM Symposium on Virtual Reality Software and Technology* (D. Thalmann, ed.), (New York, NY), ACM Press, 1997.
- [3] D. Stricker, G. Klinker, and D. Reiners, "A fast and robust line-based optical tracker for augmented reality applications," in *IWAR '98: Proceedings of the international workshop on Augmented reality : placing artificial objects in real scenes*, (Natick, MA, USA), pp. 129–145, A. K. Peters, Ltd., 1999.
- [4] U. Neumann and Y. Cho, "A selftracking augmented reality system," in *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pp. 109–115, 1996.
- [5] W. Piekarski, B. Avery, B. H. Thomas, and P. Malbezin, "Integrated head and hand tracking for indoor and outdoor augmented reality," in *VR*, pp. 11–18, 2004.
- [6] E. Foxlin and L. Naimark, "Vis-tracker: A wearable vision-inertial self-tracker," in *VR*

- '03: *Proceedings of the IEEE Virtual Reality 2003*, (Washington, DC, USA), p. 199, IEEE Computer Society, 2003.
- [7] Kato and Billinghurst, "Marker tracking and hmd calibration for a video-based augmented reality conferencing system," in *2nd International Workshop on Augmented Reality (IWAR 99)*, 1999.
- [8] D. Wagner, "Artoolkitplus," http://studierstube.org/handheld_ar, September 2005.
- [9] P. Malbezin, W. Piekarski, and B. Thomas, "Measuring artoolkit accuracy in long distance tracking experiments," in *1st Int'l AR Toolkit Workshop*, 2002.
- [10] D. F. Abawi, J. Bienwald, and R. Dorner, "Accuracy in optical tracking with fiducial markers: An accuracy function for artoolkit," in *ISMAR '04: Proceedings of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'04)*, (Washington, DC, USA), pp. 260–261, IEEE Computer Society, 2004.
- [11] R. Galantay, J. Torpus, and M. Engeli, "'living-room': interactive, space-oriented augmented reality," in *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, (New York, NY, USA), pp. 64–71, ACM Press, 2004.
- [12] J. Newman, M. Wagner, M. Bauer, A. MacWilliams, T. Pintaric, D. Beyer, D. Pustka, F. Strasser, D. Schmalstieg, and G. Klinker, "Ubiquitous tracking for augmented reality," in *ISMAR*, pp. 192–201, 2004.
- [13] M. Kalkusch, T. Lidy, M. Knapp, G. Reitmayr, H. Kaufmann, and D. Schmalstieg, "Structured visual markers for indoor pathfinding," in *The First IEEE International Augmented Reality Toolkit Workshop*, 2002.

Fast and Adaptive Block-based Motion Estimation for Video Coding

G. Sorwar¹ and M. Murshed²

¹ School of Commerce and Management, Southern Cross University, NSW 2457, Australia

² Gippsland School of Information Technology, Monash University, Vic 3842, Australia

Email: golam.sorwar@scu.edu.au

Abstract

Fast and adaptive motion estimation is still a challenge for real-time video coding applications. Previously the novel concept of a distance-dependent thresholding search (DTS) was introduced for performance scalable motion estimation in video coding applications. In this paper, the DTS algorithm has been extended to a fast and fully adaptive DTS (FFADTS) algorithm. Experimental results confirm the performance of the FFADTS algorithm in achieving better search speed over any existing fast algorithms including the diamond search (DS) and hexagon-based search (HEXBS), while maintaining similar error performance.

Keywords: Video coding, fast motion estimation, adaptive performance management, distance depending threshold.

1 Introduction

Block matching algorithm (BMA) is one of the key technologies in video compression and is widely applied in many of today's video coding standards [1-3]. The exhaustive BMA, known as the *full search* (FS) [4] algorithm, searches each candidate block for the closest match within the entire search region to minimize the *block-distortion measure* (BDM) at the expense of a very high computational overhead. It is for this reason that FS is not appropriate for any real-time video coding application.

A number of fast BMAs have been proposed to lower the computation complexity. Among them, the three-step search (TSS) [5], the new three-step search (NTSS) [6], the advanced centre biased search [7], the four-step search (FSS) [8], and recently proposed diamond search (DS) [9], and the hexagon-based search (HEXBS) [10] are mostly well known. The DS technique has achieved a significant speed gain by considering diamond-shaped search patterns instead of the conventional square ones with a view to approximate the optimal (but unrealizable) circular shape as closely as possible. Recently, the HEXBS technique has surpassed the speed of the DS technique by using a better approximation with hexagon-shaped search patterns. All of these fast algorithms suffer from the following two limitations. Firstly, they are based on the assumption that either the error surface is unimodal over the entire search area (i.e., there is only one global minimum) or the *motion vector* (MV) is centre-biased. These assumptions do not hold true for many real video sequences because of the highly non-stationary

characteristics of the video signal. Moreover, the search directions of these algorithms can be ambiguous, leading to the MV becoming entrapped in a local minimum with a resulting degradation in predictive performance. Secondly, they do not provide flexibility in controlling the performance in terms of predicted picture quality and processing time (speed).

The authors previously addressed this matter by proposing a novel fully adaptive distance-dependent thresholding search algorithm (FADTS) by introducing the concept of a distance-dependent thresholding search (DTS) algorithm for adaptive performance management motion estimation in video coding applications [11]. This paper improves the speed performance of the FADTS for real-time coding applications.

The paper is organized as follows. The original distance-dependent thresholding (DTS) algorithm is discussed in Section 2. Section 3 details the fast and fully adaptive DTS (FFADTS) algorithm. Section 4 includes both experimental results and analysis of the performance of FFADTS against some recently proposed fast motion estimation algorithms while Section 5 concludes the paper.

2 The Distance-dependent Thresholding Search (DTS) Algorithm

Definition 1 (Search Squares SS_d): The search space with maximum displacement $\pm d$, centred at pixel $p_{cx,cy}$, can be divided into $d+1$ mutually exclusive

concentric search squares SS_τ , such that a checking point at pixel $p_{x,y}$, representing MV $(x-cx, y-cy)$, is in SS_τ if and only if $\max(|x-cx|, |y-cy|) = \tau$, for all $-d+cx \leq x \leq d+cx$, $-d+cy \leq y \leq d+cy$, and $\tau = 0, 1, \dots, d$.

It can be readily verified that the number of checking points in search square SS_τ is $\begin{cases} 1, & \tau = 0 \\ 8\tau, & \tau = 1, 2, \dots, d \end{cases}$ (1)

and SS_τ represents the motion vectors of length in the range of $[\tau, \tau\sqrt{2}]$. The checking points used in the first three search squares are shown in figure 1.

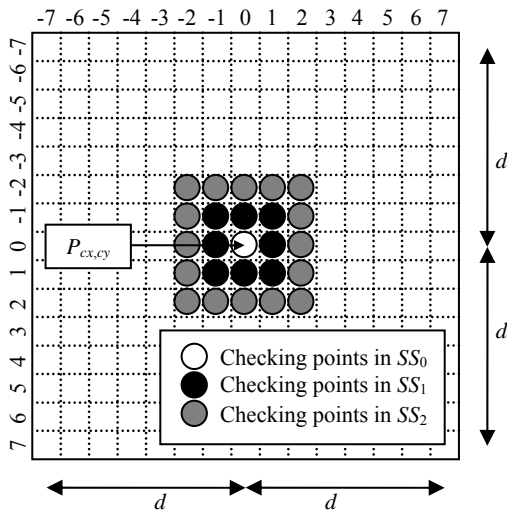


Figure 1. DTS search squares SS_0 , SS_1 , and SS_2 .

Like all block-based motion estimation search techniques, the DTS algorithm starts at the centre of the search space. The search then progresses outwards by using search squares SS_τ in order while monitoring the current minimum *mean absolute error* (MAE). A parametric thresholding function, $Threshold(\tau, C)$, is used to determine the various thresholds to be used in the search involving each SS_τ where the parameter C is set at the start of each search and acts as a control parameter. After searching each SS_τ , the current minimum MAE is compared against the threshold value of that specific search square and the search is terminated if this MAE value is not higher than that threshold value.

2.1 Characteristics of the Thresholding Function

To ensure that the DTS algorithm can be transformed to an exhaustive FS algorithm, the threshold value for SS_0 is always assumed to be 0. As the maximum MAE value using a b -bit gray level intensity is 2^b-1 , threshold values for all other search squares can, at most, be 2^b-1 . However, to ensure the algorithm includes the entire search space, all but the outermost

threshold value must be less than 2^b-1 . Moreover, to make the thresholding function distance-dependent, the function must monotonically increase. The DTS algorithm, therefore, assumes the following general properties of the thresholding function:

$$\left. \begin{aligned} Threshold(0, C) &= 0 \\ Threshold(1, C) &\leq \dots \leq Threshold(d, C) \\ Threshold(\tau, C) &< 2^b - 1 \text{ for all } \tau = 1, 2, \dots, d-1 \\ Threshold(d, C) &\leq 2^b - 1 \end{aligned} \right\} \quad (2)$$

Parameter C plays a significant role in the DTS algorithm by allowing users to define different sets of monotonically increasing threshold values based on specific values of C . Obviously, a set of larger threshold values terminates a search earlier than a set of smaller values. C , therefore, provides a control mechanism to allow trading-off between the computational complexity in terms of search points and prediction image quality.

The monotonic increasing function requirement means the DTS algorithm could use a linear, exponential, or any other complex analytic function to control the threshold with τ . In [11], the authors empirically observed linear thresholding function within the DTS algorithm outperforming and providing a wider range of flexibility compared to exponential thresholding function. This paper therefore, has considered only linear thresholding function, which is defined as follows:

$$Threshold(\tau, C_L) = C_L \times \tau, \text{ for all } \tau = 0, 1, \dots, d \quad (3)$$

The subscript L in C_L specifies linear thresholding. It can be verified that the above definition satisfies all the conditions in (2) if $C_L \geq 0$ and $C_L \times d \leq 2^b - 1$. So, parameter C_L can take any value from the range given below:

$$0 \leq C_L \leq (2^b - 1) / d. \quad (4)$$

3 Fast and Fully Adaptive DTS (FFADTS) algorithm

In adaptive motion estimation, given a target prediction image quality in terms of average *mean squared error* (MSE) per pixel, the motion search algorithm tries to achieve it using as few search checking points as possible. Inversely, if a target processing speed is set in terms of average number of search point (SP) used per MV, the algorithm tries to achieve with as low MSE as possible. Adaptive ME also assumes real time constraint, which allows very limited number of passes per macroblock. Without such a constraint, trivial trial and error technique with a very high number of passes would suffice the adaptation. Without any loss of generality, this paper assumes the strictest constraint where only one ME pass is performed per macroblock. To leverage the

adaptation technique, the original DTS algorithm is enhanced further.

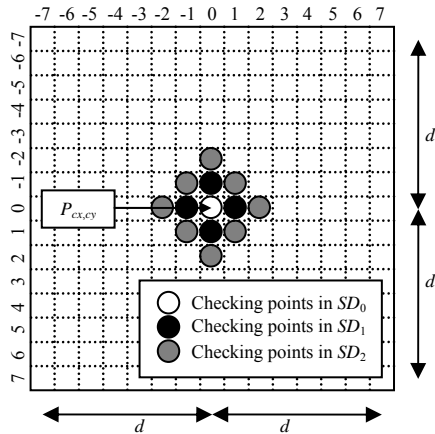


Figure 2: DTS search diamonds SD_0 , SD_1 , and SD_2 .

3.1 Enhancing the DTS Algorithm

The concept of DTS is not linked to any specific search pattern shape. In the wake of improved speed gain by non-square search patterns, DTS has been implemented using search diamonds SD_τ as shown in figure 2 where the number of checking points in SD_τ is

$$\begin{cases} 1, & \tau = 0; \\ 4\tau, & \tau = 1, 2, \dots, 2d \end{cases} \quad (5)$$

and SD_τ represents the MV of length in the range of $[\tau\sqrt{2}, \tau]$. Note that for $\tau = d + 1, d + 2, \dots, 2d$, some of the checking points in the search diamond fall outside the search windows that are obviously ignored. Using fewer checking points for centre-biased as well as horizontal and vertical motion vectors (prevalent in panning) makes DTS with search diamond superior to using search squares as observed with all the standard test sequences.

Well-established spatio-temporal motion correlation among the neighbouring macroblocks [12], [13] can be exploited to reduce the search point even further by using a predicted search origin rather than always using the centre of the search window. Assuming row-major processing order, the search origin of macroblock at r -th block row and c -th block column is calculated from the mean of the motion vectors of already processed neighbouring macroblocks at $(r-1)$ -th block row and $(c-1)$ -th block column, $(r-1)$ -th block row and c -th block column, $(r-1)$ -th block row and $(c+1)$ -th block column, and r -th block row and $(c-1)$ -th block column. If the magnitude of the difference between this mean vector with each of the four neighbouring MV is within a predefined threshold T_{pred} , the search origin at the centre is moved by that mean vector. DTS using predicted

search origin has performed superior to the original DTS for all the standard test sequences. Experimental results have also confirmed that the value of T_{pred} is not very sensitive to performance, especially for prediction error. Using the threshold in the range from 3 to 7, the average MSE and the average number of search points of the first 50 frames of *Football* and *Flower Garden* sequences varied less than 1% and 5% respectively, so to ensure average performance $T_{pred} = 5$ is defined in experiments.

3.2 The FFADTS Closed-Loop Adaptation Model

Normalized block least mean square (NBLMS) [14] can be considered as the best option for automatically adjusting the control parameter C_L in order to achieve a target average mean square error (MSE) or average SP while coding a video sequence, where this sequence can be considered as a time varying non-stationary input to the adaptation system. Based on NBLMS, the threshold control parameter is updated as:

$$C_L^{[m+K]} = C_L^{[m]} + \mu e^{[m]} \frac{\frac{1}{K} \sum_{i=0}^{K-1} y^{[m+i]}}{E_y} \quad (6a)$$

if the output is average MSE or as:

$$C_L^{[m+K]} = C_L^{[m]} - \mu e^{[m]} \frac{\frac{1}{K} \sum_{i=0}^{K-1} y^{[m+i]}}{E_y} \quad (6b)$$

if the output is the average SP where

$$E_y = \sum_{i=0}^{K-1} (y^{[m+i]})^2 \quad (7)$$

where K is block length, m is the iteration number, and μ is the step size.

4 Experimental Results

A number of experiments have been conducted to evaluate the performance of FFADTS algorithm against some recently proposed fast BMAs. Motion estimation has been carried out on the luminance (Y) values of standard video sequences *Football* (320×240 pixels, 345 frames) and *Flower Garden* (352×240 pixels, 150 frames) where both are with high object motion and camera panning respectively. For all search algorithms, block size of 16×16 pixels, and maximum search displacement of ± 7 pixels were used.

To isolate improvement due to motion search technique only, motion estimation was carried out differently than is done for video coding so that any influence of rate-distortion optimisation [15] and error propagation can be avoided. For each pair of successive frames, motion was estimated for the

second frame using the original version of the first frame (not the motion compensated version of that frame as is used for video coding) as the reference and MSE per pixel was averaged using the first frame and the motion compensated second frame. As no entropy coding was used to compress the residual, this MSE measure was higher than what could be achieved by a video coder with residual encoding. However, this MSE measure correlates highly with residual compression and thus still represents quality of the image, if rate-distortion trade off is factored in. The values of $K = 4$ and $\mu = 2$ were used in equation (6). All the search algorithms were enhanced by refining MV with half-pel accuracy using additional eight neighbouring half-pel search points (with interpolated intensity values) around the current minimum point obtained with integer-pel accuracy.

Table 1: Average MSE per pixel and sp per motion vector of the FS, TSS, NTSS, DS, and HEXBS algorithms for football and flower garden video sequences.

BMA	Football			Flower Garden		
	MSE	PSNR [dB]	SP	MSE	PSNR [dB]	SP
FS	218.9	24.7	160.1	208.9	24.9	209.7
TSS	240.8	24.3	25.6	243.0	24.3	31.2
NTSS	239.2	24.3	26.9	213.3	24.8	29.0
DS	237.0	24.4	24.9	219.7	24.7	22.8
HEXBS	241.0	24.3	21.0	226.2	24.6	20.2

Table 2: Quality adaptation for *Football* sequence.

Target Quality		Actual Quality		Actual SP
MSE	PSNR [dB]	MSE	PSNR [dB]	
230	24.51	232.04	24.48	49.18
235	24.42	234.70	24.43	32.25
240	24.32	241.00	24.31	19.87
250	24.15	252.13	24.11	16.56

Table 3: Quality adaptation for *Football* sequence.

Target Quality		Actual Quality		Actual SP
MSE	PSNR [dB]	MSE	PSNR [dB]	
210	24.91	212.80	24.85	34.95
215	24.81	214.41	24.82	24.58
220	24.71	218.62	24.73	16.65
225	24.61	222.79	24.65	15.21

Average MSE per pixel values and average search point numbers per MV for different algorithms are summarised in table 1. While FS achieves the maximum quality with the minimum average MSE

per pixel for each sequence, the speed gain of DS and HEXBS over TSS is clearly evident.

The performance of the FFADTS algorithm is shown in table 2 and 3 for quality and speed adaptation. In figure 3, quality-speed performance curves of the FFADTS algorithm are plotted for both quality and speed adaptations along with individual performance points for the TSS, NTSS, DS, and HEXBS algorithms. For the *Football* sequence (Figure 3(a)), while FFADTS outperformed TSS and NTSS in terms of quality and speed adaptations, its performance is comparable to both DS and HEXBS in both adaptations. In contrast for the *Flower Garden* sequence (Fig. 3(b)), FFADTS matches the performance of NTSS while providing superior results over all the other fast algorithms in both adaptations.

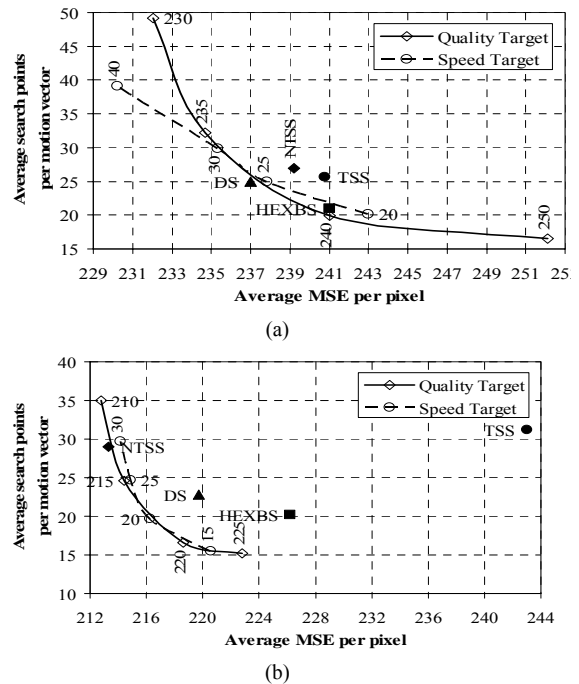


Figure 3: Quality-speed performance of the TSS, DS, HEXBS, and FFADTS algorithms for (a) *Football* and (b) *Flower Garden* video sequences. The labels on the FFADTS performance curves indicate the target values used.

In summary therefore, the FFADTS algorithm not only can adapt the threshold control parameter satisfactorily to achieve any target quality without using any more search points per MV but also any target speed with no higher MSE per pixel than the existing fast algorithms.

5 Conclusion

This paper has presented a fast and fully adaptive distance-dependent thresholding search (FFADTS) block-based motion estimation algorithm for real-time video coding applications. The search efficiency of the FFADTS algorithm has been compared to other popular fast algorithms notably the superior diamond

and hexagon-based search algorithms. Experimental results have proven that the FFADTS algorithm is not only able to provide Quality-of-Service but also demonstrates comparable or faster search speed for similar error performance and vice versa, thus addressing the problem of existing fast directional algorithms in providing different levels of quality of service.

6 References

- [1] JTC1/SC29/WG11, I.I., Generic Coding of moving pictures and associated audio. 1993, ISO/IEC.
- [2] H.263, D.I.-T.R., Video coding for low bitrate communication. 1996.
- [3] H.264, ITU-T, Infrastructure of audiovisual services – Coding of moving video, 03/2005.
- [4] J.R. Jain and A.K. Jain, “Displacement measurement and its application in inter frame image coding,” *IEEE Trans. Commun.*, vol. COM-29, pp. 1799-1808, Dec. 1984.
- [5] T. Koga, K. Iinuma, A. Hirano, Y. Iijima and T. Ishiguro, “Motion-compensated inter frame coding for videoconferencing,” *Proc. NTC81*, Nov. 1981, pp. G5. 3.1-G5.3.
- [6] R. Li, B. Zeng and M.L. Liou, “A new three-step search algorithm for block motion estimation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 438-442, Aug. 1994.
- [7] H. Nisar and T.-S. Choi, “An advanced center biased search algorithm for motion estimation,” *Proc. ICIP*, 2000, vol.1, pp.832-5.
- [8] L. M. Po and W. C. Ma, “Novel four-step search algorithm for fast block motion estimation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 313-317, June, 1996.
- [9] S. Zhu and K. -K. Ma, “A new diamond search algorithm for fast block-matching motion estimation,” *IEEE Trans. Image Processing*, vol. 9, pp. 287-290, 2000.
- [10] C. Zhu, L.-P. Chau, and X. Lin, “Hexagon-based search pattern for fast block motion estimation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 349-355, 2002.
- [11] G. Sorwar, M. Murshed, and L. Dooley, “Fast Block-based True Motion Estimation using Variable Distance Dependent Thresholds in the Full-Search Algorithm,” *Journal of Research Practice in Information Technol.*, vol. 36, no.1, pp. 83–95, February 2004.
- [12] J. -B. Xu, L.-M. Po, and C. -K. Cheung, "Adaptive motion tracking block matching algorithms for video coding," *IEEE Tran. Circuits Syst. Video Technol.*, vol. 9, pp. 1025-1029, 1999.
- [13] Y. Nam, J. -S. Seo, J. -S. Kwak, M. -H. Lee, and H. H. Yeong, "New fast-search algorithm for block matching motion estimation using temporal and spatial correlation of motion vector," *IEEE Trans. Consumer Electron.*, vol. 46, pp. 934-942, 2000.
- [14] S. C. Douglas, “A family of normalized LMS algorithms,” *IEEE Signal Processing Letters*, vol. 1, pp. 49-51, 1994.
- [15] G. J. Sullivan and T. Wiegand, “Rate-distortion optimization for video compression,” *IEEE Signal Processing Magazine*, vol. 15, pp. 74-90, 1998.

A Simple Model-Free Approach to Posture Recognition

R. Raghavan¹, K.C. Aw², S. Xie³

Department of Mechanical Engineering
University of Auckland
Auckland, New Zealand

Email: ¹rrag004@ec.auckland.ac.nz, ²k.aw@auckland.ac.nz, ³s.xie@auckland.ac.nz

Abstract

This paper presents a simple and novel approach to building a body posture classifier based on the hugely popular projection histogram technique. The intended field of application is that of visual surveillance catering to both aspects of protecting and safeguarding property or objects in museums, houses etc. It can also be expanded to detect threatening posture in video surveillance system. An algorithm for detecting postures based on asymmetry was added on to the projection histogram technique to enhance its ability to classify a larger number of posture types. With the current setup, the software developed is capable of classifying up to 11 human postures with an accuracy of 80%. The model is based on an unsupervised class set and has not been trained which allows for runtime identification and classification with no prior knowledge of the nature of the outcome. The subjects are classified based on posture matches with an exemplar set of images stored in the computers memory. A modified Manhattan distance calculator has been incorporated to compute the results of the histogram projection comparisons. The true colour images captured are binarized and down-sampled to decrease processing time whilst maintaining the same accuracy. Experiments conducted on a variety of subjects prove the validity of the model as a simple yet effective posture classifier.

Keywords: Model-free, histogram projection, human posture recognition

1 Introduction

Over the past few years, interest in the field of image processing and image classification with respect to human subjects has greatly increased. Specific topics in this field of computer vision research which have generated a large volume of research output are those of facial recognition, facial expression classification, body-posture recognition and gesture interpretation. This paper focuses on the classification of basic human postures from images captured with standard still cameras. The algorithms have been developed with a long-term goal of integrating them with other vision systems to create a self sufficient visual surveillance unit capable of detecting, analysis and classification of postures. However, this proposed model only focuses on the classification aspect of the system.

For the synthesis of a real-time model it is vital that the algorithm responsible for recognition and classification be simple and efficient enough to perform effectively without creating much of a demand for processor time. Several published journal articles point towards three basic methods of posture identification, namely, model free approaches and those based on direct or indirect models.

2 Methods of Posture Recognitions

2.1. Model Based Approaches

Pfinder and W4 are two well known examples of surveillance systems based on model based approaches. The algorithms incorporated into these two tracking systems are based on saved models where postures are identified based on the identification of specific features such as heads,

hands, legs etc. They require that the images used are of high definition and clarity, hence increase memory storage and computational time [1,2]. These models however have proven to be more effective in classifying in the occurrence of occlusions and as a result about 40% of the algorithms proposed make use of model based approaches to classification. [3]

2.2. Model Free Approaches

These approaches do not need to search for any specific feature in the image of a person for recognition or tracking purposes. Silhouettes of images have been proven to be good enough for the classification of their poses thus needing lower definition images and only boundary information of the subject for analysis. This makes algorithms based on this technique better suited for real-time applications. These images can be used to extract skeletal data for classification as suggested in [4] or can be analysed based on object shapes or the location of the centre of mass of features. Several methods have been put forth to acquire meaningful geometric data from the image blobs. For example [5] extracts stick figures from the model through axial transformations while [6] uses distance transformation for the same purpose whilst also having the flexibility to ignore unwanted regions of the body for processing to decrease processing time.

This paper describes a model free approach to posture classification based on a histogram projection analysis technique. It further develops histogram projection technique as a classifier, along with the addition of an asymmetry measurement algorithm. The paper is divided into sections which describe the pre-processing techniques used to ready the image for analysis, the analysis technique implemented and subsequently the algorithm used to classify the images.

3 System Overview

A standard 3.2 Mega pixel digital camera manufactured by Sanyo was used through the early stages of the development of the algorithm. Further into the research phase, a low-cost web-cam manufactured by AIPTEK was tested. Both of these visual capturing devices worked well for the intended purpose. However, the use of the web-cam was preferred simply because of the technique used for subject extraction which will be described at the later part of the paper. The cameras were mounted on

tripods to ensure clear images and to minimise camera movement between photographs. The photographs were taken in a 'jpeg' format at a resolution of 2048 x 1536 pixels.

The algorithms were developed, programmed and tested using MATLAB's version 7.0.1.24704 (R14) on a computer powered by a Pentium 4 processor equipped with 512 MB RAM. MATLAB's image processing toolbox was extensively used in the development of the classifier. The person images were taken against a stable background preferably monotone or with color distinctly different to the person.

3.1 Segmentation and Extraction

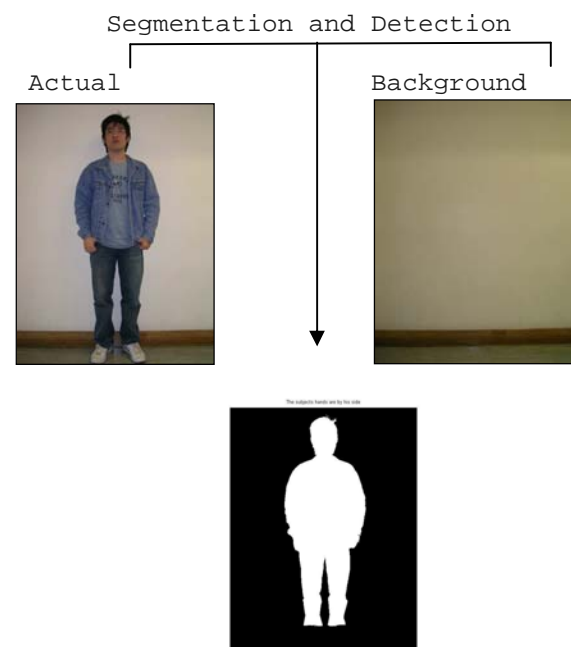


Figure 1: Segmentation and Feature Extraction

The segmentation and object detection technique is based on the fact that the application of this posture detection model is to be operated in an indoor environment where the change in background clutter is negligible and the illumination levels are relatively constant. The background frame is subtracted from the actual frame and is subsequently thresholded to produce a binary image as shown in Figure 1. Smaller noise patches which may occur due to changes in illumination or background features between successive frames are cleaned in this process by

applying a simple blob area constraint. The person's body area data is extracted and is repositioned in the centre of the frame relative to the frame to ensure accurate comparisons unaffected by translations.

3.2 Histogram Projection analysis

This is a commonly used descriptor for shape analysis. Although it has its limitation, the simplicity of the technique and also its effectiveness for simple posture analysis made it the basis of the algorithm developed for this project [7].

Since the image has been saved in a binary format, it can now be scanned along both the x and y axes to calculate the density of the pixels that are 'on', i.e. pixels with a value of 1. A histogram is developed for both the axes individually. Each histogram holds information regarding the number of 'on' pixels along each of the rows along the Y axis, in the case of the vertical projection histogram and the number of 'on' pixels along each column along the X axis in the case of the horizontal projection histogram.

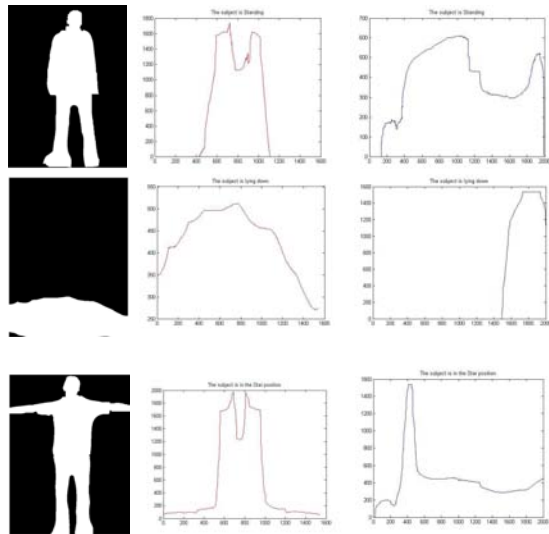


Figure 2: Horizontal (red) and vertical (blue) projection histograms for different poses

In the graphs in figure 2, the ones on the left reveal the projection data of the pixels as the photograph is scanned from left to right taking into account each column. The graphs on the right, reveal the projection data of the pixels as the photographs is scanned from top to bottom along each row.

A person who is standing will have a horizontal projection histogram quite similar to a person who is squatting and facing into the camera. However their vertical projection histograms will be drastically different. Similarly, a person lying will have a horizontal projection histogram slightly similar to a person with his hands extending perpendicularly sideways. However their vertical projection histograms are significantly different as shown in figure 2. Hence to make valid repeatable classifications based on histogram projections of body clusters, it is a must that both horizontal and vertical projections be taken into account during the analysis. Consideration of only one can easily lead the algorithm astray during classification.

3.3 Classifying asymmetric postures

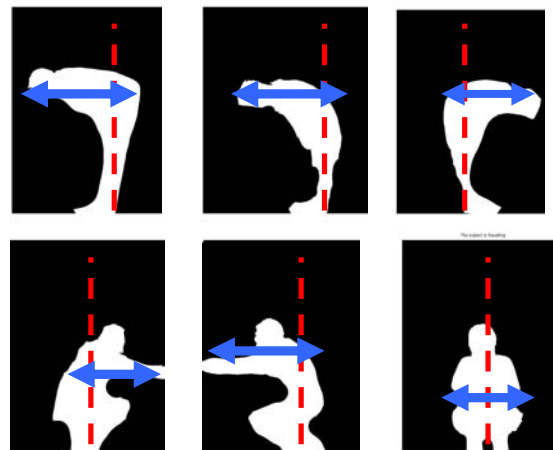


Figure 3: Identifying asymmetric postures

After observing the photographs of a large number of subjects, an interesting fact was discovered about the way humans stand and squat, a fact which has been used to develop this algorithm. It was observed that the highest value in the horizontal projection histogram occurs at a point which falls at the base of the feet of the subject. What this means is that if a vertical line was to be drawn along each column along the photograph, the line which intersected with the most number of white pixels would be located around the feet of subject. This is used as a means of locating the subject's body position in relation to the photo frame. This point, referred to as COM, can easily be identified as the values of the histogram are already stored in the memory and can be quickly scanned for a maximum value. In figure 3, the dashed red line (COM) identifies the point of maximum pixel density value and the solid line identifies the total

width of the subject's image. The solid line is used to calculate *right*, the distance to the furthest on pixel to the right of the COM and *left*, the distance of the furthest on pixel to the left of the COM. This technique is also used to check if the subject has raised either his left or right hand. A sample algorithm is as follows.

histy = sum of 'white' pixels in each column ;
COM = the column number with the most number of white pixel;
right = index of last 'white' pixel in the *histy* array ;
left = index of first 'white' pixel in *histy* array;
if $(COM-left) \div (right-COM) \geq \text{constant}$
: left hand/facing
else if $(right-COM) \div (COM-left) \geq \text{constant}$
: right hand/facing
else : symmetric

The constant is unique to the postures being analysed for example 1.3 is used while identifying the hand raised, 1.2 for squat direction and 1.2 for bend direction.

Leaving a minimum margin of 0.1 allows for small amounts of unintended or accidental asymmetry to be ignored. Experiments have lead to the discovery that the constants stated above give the best results.

4 Classifying the postures

Projection histograms are highly representative of the body posture they are developed from. Hence comparing the vertical and horizontal projection histograms of two postures can provide enough evidence to differentiate between them, provided the two postures are distinct enough. Simple distance calculations can be used to derive numbers representing the similarities or dissimilarities of histogram projections. Decisions on classifications are made by the computer based on these distance measurement results.

Two types of distance calculation techniques were considered for this purpose. One being the Euclidean distance and the other is Manhattan distance. The Euclidean distance is the straight line distance from point 'a' to 'b' whereas its Manhattan distance is the total distance from 'a' to 'b' if a grid like path was to be followed. The equations below help describe the two ideologies.

$$D_{manhtnan} = \sum_{i=1}^n \{ | (x_{i+1} - x_i) | + | (y_{i+1} - y_i) | \}$$

$$D_{Euclidean} = \sum_{i=1}^n \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$$

Although the Euclidean distance ($D_{Euclidean}$) provides more accurate measurements it was not preferred as it increases computational time, the minimisation of which is highly essential to allow for the programme to function as a real time system. It has also been proven to be more accurate for use with classification base on projection histograms [7]. The Manhattan distance ($D_{manhattan}$) between two points is calculated much quicker as the horizontal and vertical projection data are already stored as separate vectors in the memory and makes it easier to calculate the similarities between the two horizontal projections and two vertical projections individually. When the histograms of the unknown pose are compared with those of a known one, a small distance measurement indicates a closer resemblance. A variation of the Manhattan distance was implemented to calculate this distance value.

For a photograph of size 2048 x 1536 pixels, the horizontal projection vector would contain 1536 values and the vertical projection vector 2048 values. Each value of each vector is compared with its counterpart from the prototype image's vectors. Thus it is vital that the photos stored in the database and those taken be of the same dimensions. The following steps were used to calculate histogram similarities between image one, I1 and image two, I2.

$$D(Y1, Y2) = \min \left(\sum_{i=1}^{DimY} | Y1_j - Y2_{j+1} | ; \sum_{i=1}^{DimY} | Y2_j - Y1_{j+1} | \right),$$

$$D(X1, X2) = \min \left(\sum_{j=1}^{DimX} | X1_j - X2_{j+1} | ; \sum_{j=1}^{DimX} | X2_j - X1_{j+1} | \right),$$

$$\sum_{i=1}^{DimX} | X1_j - X2_{(DimX-j-1)} | ; \sum_{i=1}^{DimX} | X2_j - X1_{(DimX-j-1)} |$$

where

$D(Y1, Y2)$ = least mean error between the two postures in y-direction.

$D(X1, X2)$ = least mean error between the two posturise in x-direction.

DimY and DimX = the maximum size of the image in Y and X direction respectively.

The additional terms in the calculations for similarities between the horizontal projection data help iron out differences which may occur due to slight translation or mirroring effects about the

vertical axis. Mirroring about the horizontal axis is highly unlikely [3].

The overall similarity (least difference between two images) between the two projection histograms, i.e. taking both into account, can be calculated as following.

$$D(Im1,Im2) = a * D(Y1,Y2) + b * D(X1,X2)$$

The variables ‘a’ and ‘b’ refer to weights which may be given to the individual histogram similarities. Several experimental tests have revealed that unless specifically intended both histograms must be given equal weighting for ‘a’ and ‘b’.

To detect the similarity of a pose in comparison to two known poses the following mathematical calculation is performed to provide a percentage resemblance [3].

$$D(Im,STAND)= 100 - \left\{ \frac{D(Im,STAND)}{D(Im,STAND) + D(Im,LYING)} \right\} * 100$$

This expression gives a result as a percentage which indicates the likely hood of the pose being a stand rather than a lie. It was found through experimentation that the more distinct the two poses under comparison, the more reliable the result. The likelihood of the posture being a ‘lie’ is simply a complement of a ‘stand’.

Three different subjects were chosen to construct the template database for the various postures. To cover a large range of images of body sizes of a tall, short and an average male were taken in all the postures to create the database. Histogram templates were created and stored for comparisons during the analysis of the unidentified postures in the captured images. Unlike techniques such as contour based descriptors and shape context matching, this technique does not demand a large template set. However it is necessary to cover the extremes of height and width ranges of the possible subjects to improve the chances of a successful classification. The three sets of histogram data are averaged out and saved as a template (figure 4). There was no need to capture images of persons with varying widths for the horizontal projection templates as these histograms do not play a major role in the classification of postures. The method of analysis chosen is more heavily depended on vertical histogram data.

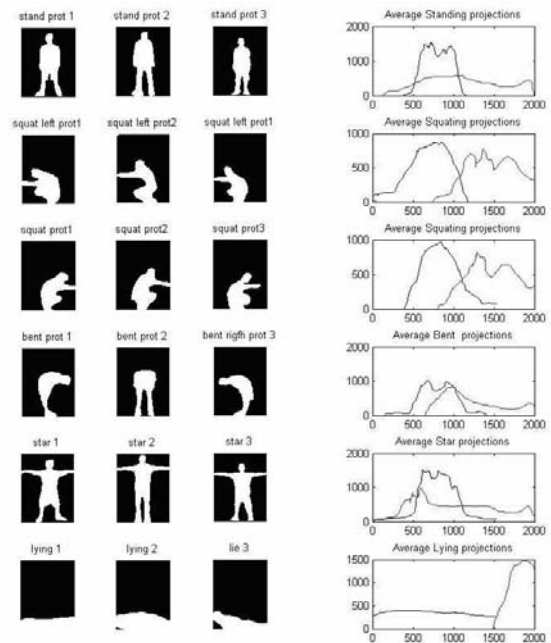


Figure 4: Averaged comparison templates (red denotes vertical histogram)

The classification process in can be divided into 3 main steps as follows.

Step 1 - Check for ‘stand’ or ‘lie’ posture

A comparison is made between the similarity of the subject’s histogram with those of the predefined standing and lying postures. During this phase a higher weighting is given to the vertical projection histogram as it is this aspect which possesses greatest diversity between the two postures. Decisions are made on the results from this comparison

Step 2 – Check for ‘squat’ or ‘bend’ posture

The image is analysed to check if the subject is squatting or bent. Projection histograms are used again for this classification step taking into consideration only the vertical projection histogram. It has been observed that a person squatting always produces a shorter profile than a person who is bent. Hence this fact is utilised to identify the two postures.

Step 3 – Asymmetry analysis for further classification

The asymmetry detection algorithm explained earlier is used to identify the person’s hand raised and the orientation.

Several experimental runs have identified the optimised constants for best recognition percentage

5 Experimental results

The results were summarised in Figure 5. In the figure, different postures with their corresponding recognition percentages are presented. The posture that reports the highest percentage would denote that the posture of the capture image is most likely. For example, if an image has 72.83% for bent and 27.17% for squat, it would suggest that the image posture would likely be a 'bent'. The direction denotes the direction the person in the image is facing.

6 Discussions and Conclusion

The classification approach used has been proven to classify up to 11 postures with a reasonable accuracy of about 80%. Although histogram projections of an object are calculated relative to the frame of view, this algorithm manages to avoid sensitivity to the object's position in the frame of view.

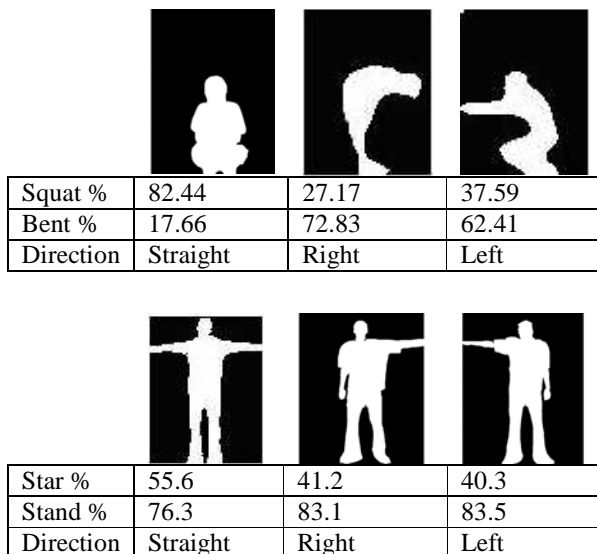


Figure 5: Test results for some posture classifications

This is done by generating histograms for classification only after centring the subject in the frame after pre-processing. Also the histogram data generated is smoothed by a weighting factor which has been optimised after several trials. This

smoothing provides error compensation which might be caused due to lateral positioning of the subject.

The success of this classification algorithm is highly dependent on how well the images were pre-processed. However, for a successful pre-processing, it is important that the image background is clutter free and the subject is not occluded by other object.

7 References

- [1] I. Haritaoglu, D. Harwood, L.S. Davis, "W4: real-time surveillance of people and their activities", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 22(8), 2000, pp.809-830.
- [2] C.R. Wren, A. Azarbayejani, T. Darrell, A. P. Pentland, "Pfinder: real-time tracking of the human body", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 19(7), 1997, pp. 780-785,.
- [3] P. Spagnolo, M. Leo, G. Attolico, A. Distanto, "Posture Recognition in Visual Surveillance of Archaeological Sites", *IEEE Proceedings Conference on Intelligent Robots and Systems*, October 2003.
- [4] C. Castiello, T. D'Orazio†, A. M. Fanelli, P. Spagnolo, M. A. Torsello, "A Model-free approach for posture classification", *Proceedings. IEEE Conf. on Adv. Video and Signal Based Surveillance*, 2005. Sept. 2005, pp276 - 281
- [5] A.G.Bharatkumar, KE.Daigle, M.G.Pandy, Q. Cai, J.K.Agganval, "Lower limb kinematics of human walking with the medial axis transformation", in *Proc. of the Workshop on Motion of Non-Rigid and Articulated Objects*, Austin, TX, , November 1994, pp. 70-76.
- [6] S. Iwasawa, K. Ebihara, J. Ohya, S. Morisbima, "Realtime estimation of human body posture from monocular thermal images", in *Proc. of 1999 IEEE Computer Society Conz on Comp. Vision and Patt. Recogn.*, 1997, pp.15-20,.
- [7] Lutz Goldmann, Mustafa Karaman & Thomas Sikora, "Human body posture recognition using MPEG-7 Descriptors". *Proceedings of SPIE Visual Communications and Image Processing 2004*, Vol. 5308, Jan. 2004, pp. 177-188.

Genetic Programming for Object Detection

Mengjie Zhang, Urvesh Bhowan, Bunna Ny

School of Mathematics, Statistics and Computer Science,
Victoria University of Wellington, PO Box 600, Wellington, New Zealand

Email: {mengjie,urvesh,bunna}@mcs.vuw.ac.nz

Abstract

This paper describes a genetic programming approach to object detection. This approach breaks the GP search into two phases with the first phase applied to a selected subset of the training data, and a simplified fitness function. The second phase is initialised with the programs obtained from the first phase, and uses the full set of training data with a complete fitness function to construct the final detection programs. In addition to the detection rate and false alarm rate, a program size and a false alarm area components are added to the fitness function. The results on two object detection problems suggest that the proposed approach improve the effectiveness and the efficiency of genetic programming.

Keywords: Artificial Intelligence approaches to Computer Vision, Object Recognition, Image Analysis, Genetic Programming, Neural Networks.

1 Introduction

Object detection tasks arise in a very wide range of applications, such as detecting faces from video images, finding tumours in a database of x-ray images, and detecting cyclones in a database of satellite images. In many cases, people (possibly highly trained experts) are able to perform the classification task well, but there is either a shortage of such experts, or the cost of people is too high. Given the amount of data that needs to be detected, computer based object detection systems are of immense social and economic value.

An object detection program must automatically and correctly determine whether an input vector describing a portion of a large image at a particular location in the large image contain an object of interest or not and what class the suspected object belongs to. Writing such programs is usually difficult and often infeasible: human programmers often cannot identify all the subtle conditions needed to distinguish between all objects and background instances of different classes.

Genetic programming (GP) is a relatively recent and fast developing approach to automatic programming [1, 2]. In GP, solutions to a problem are represented as computer programs. Darwinian principles of natural selection and recombination are used to evolve a population of programs towards an effective solution to specific problems.

There have been a number of reports on the use of GP in object detection [3, 4, 5, 6, 7, 8, 9]. The approach we have used in previous work [8, 9] is to use a single stage approach (referred to as *the*

basic GP approach here), where the GP is directly applied to the large images in a moving window fashion to locate the objects of interest. Past work has demonstrated the effectiveness of this approach on several object detection tasks.

While showing promise, the GP approach still has some problems. One problem is that the training time was often very long, even for relatively simple object detection problems. A second problem is that the evolved programs are often hard to understand or interpret. The big size of the programs with redundancy contributes to the long training times. Evaluating the fitness of a candidate detector program in the basic GP approach involves applying the program to each possible position of a window on all the training images, which is quite expensive.

The goal of this paper is to investigate two ideas that can improve the above two situations. The first is to split the GP evolution into two phases, using a simple fitness function and just a subset of the training data in the first phase. The second idea is to augment the fitness function in the second phase by a component that biases the evolution towards smaller, less redundant programs. We consider the effectiveness and efficiency of this approach by comparing it with the basic GP approach and a neural network approach.

The rest of the paper is organised as follows. Section 2 presents the main aspects of this approach. Section 3 describes the three image data sets and section 4 presents the experimental results. Section 5 draws the conclusions and gives future directions.

2 GP Adapted to Object Detection

The term *object detection* here refers to the detection of small objects in large images. This includes both *object classification* and *object localisation*. *Object classification* refers to the task of discriminating between images of different kinds of objects, where each image contains only one of the objects of interest. *Object localisation* refers to the task of identifying the positions of all objects of interest in a large image.

Object detection performance is usually measured by *detection rate* and *false alarm rate*. The detection rate (DR) refers to the number of small objects correctly reported by a detection system as a percentage of the total number of actual objects in the image(s). The false alarm rate (FAR), also called false alarms per object [10], refers to the number of non-objects incorrectly reported as objects by a detection system as a percentage of the total number of actual objects in the image(s). Note that the detection rate is between 0 and 100%, while the false alarm rate may be greater than 100% for difficult object detection problems.

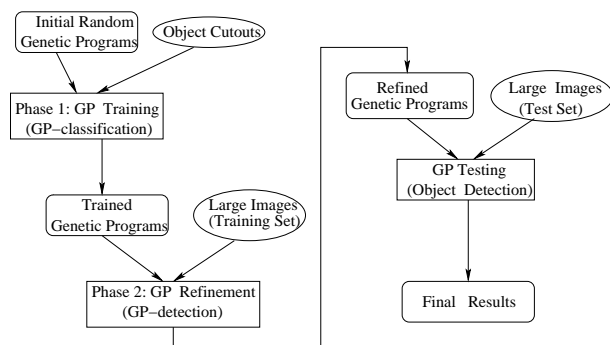


Figure 1: An overview of the approach.

Figure 1 shows an overview of this approach, which has two phases of learning and a testing procedure. In the first learning phase, the evolved genetic programs were initialised randomly and trained on object examples cut out from the large images in the training set. This is just an object classification task, which is simpler than the full object detection task. This phase therefore uses a fitness function which maximises classification accuracy on the object cutouts. In the second phase, a second GP process is initialised with the programs generated by the first phase, and trained on the full images in the training set by applying the programs to a square input field (“window”) that was moved across the images to detect the objects of interest. This phase uses a fitness function that maximises *detection* performance on the large images in the training set. In the test procedure, the best refined genetic program is then applied to the entire

images in the test set to measure object detection performance.

Because the object classification task is simpler than the object detection task, we expect the first phase to be able to find good genetic programs much more rapidly than the second phase. Although simpler, the object classification task is closely related to the detection task, so we expect the genetic programs generated by the first phase to be very good starting points for the second phase.

Since the difficulty of finding an optimal program increases with the size of the programs, in the second phase, we include a program size component to the fitness function to bias the search towards simpler programs. We expect this to improve the system accuracy and efficiency, and also make the programs easier to interpret.

Notice that the two phase approach here will only produce a *single program* for the whole detection task. This is different from the typical multi-stage approach with a program/system for each stage. In the rest of the section, we will describe the main aspects of the GP system, including the primitive set, the fitness function, the parameters, and termination criteria of the evolutionary process.

2.1 Primitive Sets

For object detection problems, terminals generally correspond to image features. Instead of using global features of an entire input image window, we used a number of statistical properties of local square and circular region features as terminals, as shown in figure 2. The first terminal set consists of the means and standard deviations of a series of concentric square regions centred in the input image window, which was used in the *shape* data set (see section 3). The second terminal set consists of the means and standard deviations of a series of concentric circular regions, which was used in the *coin* data set. Notice that these features are certainly not the best for these particular problems, however, our goal is to investigate the two-phase and the program size ideas rather than finding good features for a particular task. We also added some random constants to each terminal set.

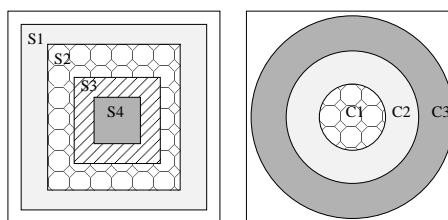


Figure 2: Local square and circular features.

In the function set, the four standard arithmetic operators and a conditional operator were used to form the non-terminal nodes: $\text{FuncSet} = \{+, -, *, /, \text{if}\}$. The $+$, $-$, and $*$ operators have their usual meanings — addition, subtraction and multiplication, while $/$ represents “protected” division which is the usual division operator except that a divide by zero gives a result of zero. Each of these functions takes two arguments. The if function takes three arguments. The first argument, which can be any expression, constitutes the condition. If the first argument is positive, the if function returns its second argument; otherwise, it returns its third argument.

2.2 Converting Programs to Classes

The output of a genetic program is a floating point number. For object detection problems, this value must be converted to a class label. In this approach, we used a variant *program classification map*, as shown in equation 1, for this purpose [11].

$$\text{Class} = \begin{cases} \text{background,} & v \leq 0 \\ \text{class 1,} & 0 < v \leq T \\ \text{class 2,} & T < v \leq 2T \\ \dots & \dots \\ \text{class } i, & (i-1) \times T < v \leq i \times T \\ \dots & \dots \\ \text{class } m, & v > (m-1) \times T \end{cases} \quad (1)$$

where m refers to the number of object classes of interest, v is the output value of the evolved program and T is a constant defined by the user, which plays a role of a threshold.

2.3 Fitness Functions

As mentioned earlier, we used two fitness functions for the two learning phases. The first phase used the classification accuracy directly as the fitness function to maximise object classification accuracy. The second phase used a *multi-objective* fitness function to maximise object detection accuracy, which is to be described below.

The goal of object detection is to achieve both a high detection rate and a low false alarm rate. In genetic programming, this typically needs a multi-objective fitness function. A fitness function we used in previous work [9] is:

$$\text{fitness}(DR, FAR) = W_d * (1 - DR) + W_f * FAR \quad (2)$$

where DR is the Detection Rate and FAR is the False Alarm Rate, as described earlier. The parameters W_d, W_f reflect the relative importance between the detection rate and the false alarm rate.

Although such a fitness function accurately reflects the performance measure of an object detection system, it is not smooth. In particular, small improvements in an evolved genetic program may not be reflected in any change to the fitness function. The reason is the clustering process that is essential for the object detection — as the sliding window is moved over a true object, the program will generally identify an object at a cluster of window locations where the object is approximately centred in the window. It is important that the set of positions is clustered into the identification of a single object rather than the identification of a set of objects on top of each other.

A poor program may produce a larger cluster of “incorrect” locations and a better program may produce a smaller cluster of locations (as shown in figures 3 (b) and (c)). Although the second program is better than the first, it has exactly the same FAR since both programs have two false positives. A fitness function based solely on DR and FAR cannot correctly rank these two programs, which means that the evolutionary process will have difficulty for selecting better programs. To deal with this problem, the False Alarm Area (FAA, the number of false alarm pixels which are not object centres but are incorrectly reported as object centres before clustering) was added to the fitness function.

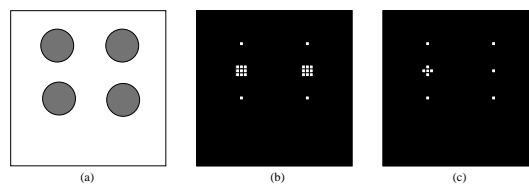


Figure 3: Sample object detection maps. (a) Original image; (b) Detection map produced by a poor program; (c) Detection map produced by a better program.

Another problem with this fitness function is that some genetic programs evolved are often very long. When a short program and a long program produce the same detection rate and the same false alarm rate, the GP system will randomly choose one for reproduction, mutation or crossover during the evolutionary process. If the long programs are selected, the evolution for the rest of the learning process will be slow. This is mainly because this fitness function does not include any direct heuristics about the size of programs.

2.3.1 The New Fitness Function

To smooth the fitness function so that small improvement in genetic programs could be reflected and to consider the effect of program size, we added

two measures, *false alarm area* and *program size* to the fitness function.

The fitness of a genetic program in the new fitness function is calculated as follows.

1. Apply the program as a moving $n \times n$ window template (n is the size of the input image window) to each of the training images and obtain the output value of the program at each possible window position. Label each window position with the ‘detected’ object according to the object classification strategy. Call this data structure a detection map.
2. Find the centres of *objects of interest only* by the clustering algorithm:
 - Scan the detection map for an object of interest. When one is found mark this point as the centre of the object and continue the scan. Skip pixels in $n/2 \times n/2$ square to right and below this point.
3. Match these detected objects with the known locations of each of the desired true objects and their classes.
4. Calculate the detection rate DR , the false alarm rate FAR , and the false alarm position FAA of the evolved program.
5. Count the size of the program by adding the number of terminals and the number of functions in the program.
6. Compute the fitness of the program according to equation 3.

$$fitness = K_1 \cdot (1 - DR) + K_2 \cdot FAR + K_3 \cdot FAA + K_4 \cdot ProgSize \quad (3)$$

where K_1, K_2, K_3 , and K_4 are constant weighting parameters which reflect the relative importance between DR, FAR, FAA , and the program size.

We expect the new fitness function to reflect both small and large improvements of the genetic programs, bias the search towards simpler functions, and accordingly to improve both the efficiency and the effectiveness of the evolutionary search. It will also have a tendency to reduce redundancy, making the programs more comprehensible.

2.4 Parameters and Termination Criteria

In this system, we used tree structures and Lisp S-expressions to represent genetic programs [2]. The ramped half-and-half method [1, 2] was used for generating the programs in the initial population and for the mutation operator. The proportional selection mechanism and the reproduction [11], crossover and mutation operators [1] were used in the learning process.

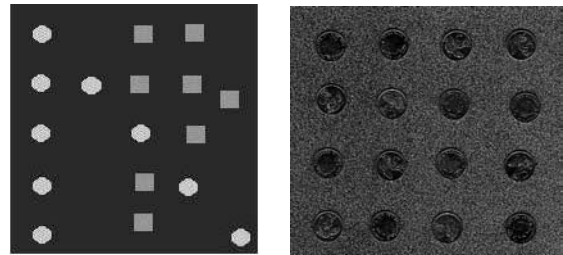
Parameter values used in this approach are shown in table 1. The learning process is run for a fixed number (*max-generations*) of generations, unless it finds a program that solves the problem perfectly, or there is no increase in the fitness for 10 generations, at which point the evolution is terminated early.

Table 1: Parameters used for GP training.

Parameters	Parameter Name	Shape	Coins	Heads/tails
Search Parameters	population-size	800	1000	1600
	initial-max-depth	2	2	5
	max-depth	6	7	8
	max-generations	50	150	200
	input-size	20×20	72×72	62×62
Genetic Parameters	reproduction-rate	2%	2%	2%
	cross-rate	70%	70%	70%
	mutation-rate	28%	28%	28%
Fitness Parameters	T	100	80	80
	K1	5000	5000	5000
	K2	100	100	100
	K3	10	10	10
	K4	1	1	1

3 Image Data Sets

We used two data sets in the experiments. Example images are given in figure 4. Data set 1 (Shape) was generated to give well defined objects against a uniform background. The pixels of the objects were generated using a Gaussian generator with different means and variances for different classes. There are two classes of small objects of interest in this database: circles and squares. In data set 2 (coin), the task is detecting the head side and the tail side of scanned New Zealand 5 cent coins with various orientations from a cluttered background. Given the low resolution of the images, this detection task is actually very difficult — even humans cannot distinguish these objects perfectly.



No. of images: 10
Object size: 18×18
(Shape)

No. of images: 20
Object size: 60×60
(Coin)

Figure 4: Object detection problems.

In the experiments, we used one and five images as the training set and used five and ten images as the test set for the *Shape* and *Coin* data sets, respectively.

4 Results and Discussion

4.1 Effectiveness: Detection Accuracy

To investigate the performance of this approach, we compared this approach with the basic GP approach [8, 12] and a neural network approach [13, 14] using the same set of features. The basic GP approach is similar to the approach described in this paper, except that it uses the old fitness function without considering the program size and false alarm areas (equation 2) and that genetic programs are learned from the full training images directly, which is a single stage approach. In the neural network approach [13, 14], a three layered feed forward neural network is trained by the back propagation algorithm [16] without momentum using an online learning scheme and fan-in factors. For all the three approaches, the experiments are repeated 50 times and the average results on the *test set* are presented in this section.

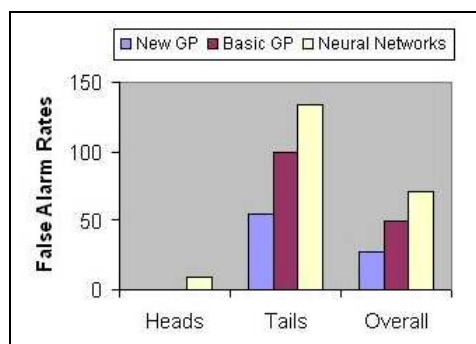


Figure 5: Object detection results.

All the three approaches achieved the ideal results for the *Shape* data set, reflecting the fact that the detection problem in this data set is relatively easy. For the difficult *Coin* data set, none of the three methods resulted in ideal performance. While all the three approaches also achieved 100% detection rate for the *Coin* data set, they produced very different false alarm rates. The false alarm rates for the two classes and the overall task are presented in figure 5. The results suggest that the new two-phase GP approach described in this paper achieved the best performance and that both GP approaches achieved better results than the neural network approach using the same set of features.

4.2 Efficiency

Although both of the GP approaches achieved better results than the neural networks overall, the times spent on the training/refining process are quite different. For the *Coin* data set, for example, the two phase GP approach found good programs after 23 hours on average, while the basic GP approach took an average of 45 hours. The first phase

is so fast because the size of the training data set is small, and the task of discriminating the classes of objects (when centred in the input window) is relatively simple. However, the programs it finds appear to be very good starting points for the more expensive second phase, which enables the evolution in the second phase to concentrate its search in a much more promising part of the search space.

The execution times of the two GP approaches on the test sets are only a few seconds, which is much shorter than the neural network approach. There are two main reasons. Firstly, the functions in the best evolved genetic programs are simpler than those in the trained neural networks such as complex transfer functions. Secondly, while the neural networks must use all the features in the terminal set, the GP approach only selects those relevant to a particular task and makes the evolved programs more concise.

4.3 Example Evolved Programs

To check the effectiveness of the new fitness function at improving the comprehensibility of the programs, an evolved genetic program in the *shape* data set is shown below:

```
(/ (if (/ (- F4μ T) F4μ)
      F3μ
      (* (- F4μ F2μ) F1σ))
  (/ F4μ F4μ))
```

This program detector can be simplified as follows:

```
(if (- F4μ T) F3μ (* (- F4μ F2μ) F1σ))
```

where $F_{iμ}$ and $F_{iσ}$ are the mean and standard deviation of region i (see figure 2, left) of the window, respectively, and T is a predefined threshold. This program can be translated into the following rule:

```
if (F4μ > T) then
  value = F3μ;
else
  value = (F4μ - F2μ) * F1σ;
```

If the sweeping window is over the background only, $F_{4μ}$ would be smaller than the threshold (100 here), the program would execute the “else” part. Since $F_{4μ}$ is equal to $F_{2μ}$ in this case, the program output will be zero. According to the classification strategy — object classification map, this case would be correctly classified as *background*. If the input window contains a portion of an object of interest and some background, $F_{4μ}$ would be smaller than $F_{2μ}$, which results in a negative program output, corresponding to class *background*. If

$F_{4\mu}$ is greater than the threshold T , then the input window must contain an object of interest, either for *class1* or for *class2*, depending the value of $F_{3\mu}$.

While this program detector can be relatively easily interpreted, the programs obtained using the old fitness function are generally hard to interpret due to the length of the programs. The trained neural networks, with many links and a complex transfer function at each node, are almost a “black box”.

5 Conclusions

The paper investigated a two phase GP approach with a new fitness function including a program size and a false alarm area components. Our results suggest that the two phase approach is more effective and more efficient than the basic GP approach and more effective than a neural network approach on the two data sets using the same set of features. The modified fitness function resulted in genetic program detectors that were better quality and easier to interpret.

While this approach considerably shortens the training times, the training process is still relatively long. We will explore better classification strategies and add more heuristics to the genetic beam search to the evolutionary process in the future.

References

- [1] W. Banzhaf, P. Nordin, R. E. Keller, and F. D. Francone, *Genetic Programming: An Introduction on the Automatic Evolution of computer programs and its Applications*. Morgan Kaufmann Publishers, 1998.
- [2] J. R. Koza, *Genetic programming : on the programming of computers by means of natural selection*. MIT Press, 1992.
- [3] W. A. Tackett, “Genetic programming for feature discovery and image discrimination,” in *Proceedings of the 5th International Conference on Genetic Algorithms, ICGA-93* (S. Forrest, ed.), pp. 303–309, Morgan Kaufmann, 17-21 July 1993.
- [4] K. Benson, “Evolving finite state machines with embedded genetic programming for automatic target detection within SAR imagery,” in *Proceedings of the 2000 Congress on Evolutionary Computation CEC00*, (USA), pp. 1543–1549, IEEE Press, 6-9 July 2000.
- [5] C. T. M. Graae, P. Nordin, and M. Nordahl, “Stereoscopic vision for a humanoid robot using genetic programming,” in *Real-World Applications of Evolutionary Computing* (S. Cagnoni, et al. eds.), vol. 1803 of *LNCS*, (Edinburgh), pp. 12–21, Springer-Verlag, 17 Apr. 2000.
- [6] D. Howard, S. C. Roberts, and C. Ryan, “The boru data crawler for object detection tasks in machine vision,” in *Applications of Evolutionary Computing* (S. Cagnoni, et al. eds.), vol. 2279 of *LNCS*, (Kinsale, Ireland), pp. 220–230, Springer-Verlag, 3-4 Apr. 2002.
- [7] F. Lindblad, P. Nordin, and K. Wolff, “Evolving 3d model interpretation of images using graphics hardware,” in *Proceedings of the 2002 IEEE Congress on Evolutionary Computation, CEC2002*, (Honolulu, Hawaii), 2002.
- [8] M. Zhang and V. Ciesielski, “Genetic programming for multiple class object detection,” in *Proceedings of the 12th Australian Joint Conference on Artificial Intelligence, LNAI Volume 1747*. (N. Foo, ed.), pp. 180–192, Springer-Verlag, 1999.
- [9] M. Zhang, P. Andreae, and M. Pritchard, “Pixel statistics and false alarm area in genetic programming for object detection,” in *Applications of Evolutionary Computing, Lecture Notes in Computer Science, LNCS Vol. 2611* (S. Cagnoni, ed.), pp. 455–466, Springer-Verlag, 2003.
- [10] M. V. Shirvaikar and M. M. Trivedi, “A network filter to detect small targets in high clutter backgrounds,” *IEEE Transactions on Neural Networks*, vol. 6, pp. 252–257, Jan 1995.
- [11] M. Zhang, V. Ciesielski, and P. Andreae, “A domain independent window-approach to multiclass object detection using genetic programming,” *EURASIP Journal on Signal Processing*, vol. 2003, no. 8, pp. 841–859, 2003.
- [12] U. Bhowan, “A domain independent approach to multi-class object detection using genetic programming,” BSc Honours research project, School of Mathematical and Computing Sciences, Victoria University of Wellington, 2003.
- [13] M. Zhang and V. Ciesielski, “Using back propagation algorithm and genetic algorithm to train and refine neural networks for object detection,” in *Proceedings of the 10th International Conference on Database and Expert Systems Applications, LNCS Volume 1677*. pp. 626–635, Springer-Verlag, 1999.
- [14] B. Ny, “Multi-class object classification and detection using neural networks,” BSc Honours research project, School of Mathematical and Computing Sciences, Victoria University of Wellington, 2003.
- [15] M. Zhang, P. Andreae, and U. Bhowan. “A two phase genetic programming approach to object detection”, in *Lecture Notes in Artificial Intelligence, Vol. 3215 (KES04, Part III)*. pp 224-231, 2004.
- [16] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning internal representations by error propagation,” in *Parallel distributed Processing, Explorations in the Microstructure of Cognition, Volume 1: Foundations* ch. 8, The MIT Press, 1986.

Object Indexing and Recognition

F. Souami, S. Aouat

LRIA, Computer Department, University of Science and Technology – Houari Boumediene

Email: fsouami@usthb.dz

Abstract

In this paper, we address the problem of recognizing 3D object from a single image using objects models database. We used geometric quasi-invariant features issued from images to build up the database. These features sets are the object indexes. To code these indexes, and to enhance the recognition process, we propose a modified X-tree technique. On the other hand, to overcome high database dimensionality retrieval difficulties, we introduce a vector approximation file to transform the indexes space into a similarity space. Distance between image and database models indexes are calculated in the similarity space. Our final vote method is used to reject not matching objects within the database.

Keywords: object database, geometric quasi invariant, content based, similarity, indexing, retrieval.

1 Introduction

An easy way to recognize an object in an image is to find object with best resemblance in the database [1, 2]. This problem can be considered as an indexing retrieval problem which consists in index calculation from a set of image features, and comparison with object database indexes. Several image features can compose object index. We have used geometric features.

3D object indexing problem is the purpose of a large number of research work [3, 4, 5]. Segments are interesting features because of their robustness to noise and their connectedness constraint (based on a topological reality in the image). They also have the properties to vary slightly with a small change in the viewpoint, and to be invariant under similarity transform of the image [6].

We used geometric invariant features to match objects. These features are geometric quasi-invariants (ρ, θ) defined by intersecting segments [7]. These features are also used as object indexes.

The object indexing retrieval system we propose is based on geometric quasi invariant indexes. Instead of interpreting 3D information, we perform the object indexing and retrieval in 2D index space (figure 1).

The X-tree algorithm has been adopted to code these indexes instead of the hashing table. This creates high dimensionality database. The vector approximation file technique [6] provides a method to overcome the high dimensionality curse, by following not the data partitioning approaches of conventional index methods, but rather act as a filter based approach.

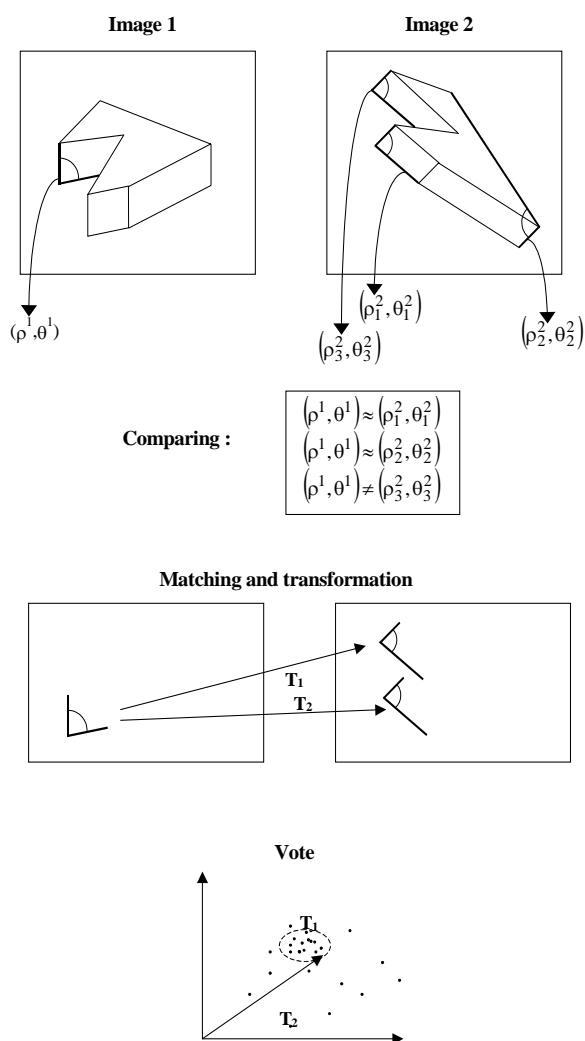


Figure 1. Object indexing retrieval system.

To recognize an object in a request image, we propose a three step method. First, (ρ, θ) features are extracted from image and geometric indexes built. Second, the indexes are coded in a X-tree and then in a similarity space by the VA file technique. Third, distance between the image request indexes and the object database indexes are calculated in the similarity space. A final vote step identifies the best matching object from the object database. Once the object is recognised, it is easy to back track the 3D information.

2 Geometric Indexes

2.1 Quasi-invariants and Similarity

The quasi-invariants (ρ, θ) are the angle θ between intersecting segments, and segments length ratio ρ [7].

$$\rho = \frac{\overrightarrow{a_0 a_1}}{\overrightarrow{a_0 a_2}} \quad \theta = \arccos \frac{\overrightarrow{a_0 a_1} \cdot \overrightarrow{a_0 a_2}}{\|\overrightarrow{a_0 a_1}\| \|\overrightarrow{a_0 a_2}\|}$$

Considering two geometric configurations (figure 2), similarity is expressed as homothety k , rotation α , and translation \vec{T} :

$$k = \frac{1}{2} \left(\frac{l'_1}{l_1} + \frac{l'_2}{l_2} \right)$$

$$\alpha = |\theta'_0 - \theta_0| + \frac{1}{2} (\theta' - \theta)$$

$$T = \begin{pmatrix} x_b - k(x_a \cos \alpha - y_a \sin \alpha) \\ y_b - k(x_a \sin \alpha + y_a \cos \alpha) \end{pmatrix}$$

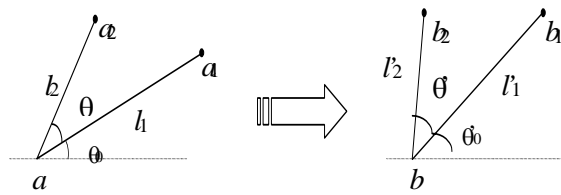


Figure 2. Geometric quasi invariant and similarity.

2.2 Image Features

We propose the following image features: quasi invariant (ρ, θ) , colour RGB, intersecting segments lengths (l_1, l_2) and the image identifier (number).

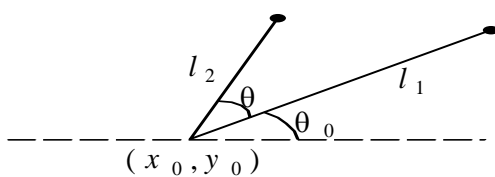


Figure 3. Index composition.

2.3 Geometric Features analysis

In order to build the indexing structure, we first analyzed the (θ, ρ) distribution. Figure 4 shows that $\pi/2$ is the θ distribution most frequent value. Figure 5 shows a non uniform ρ distribution.

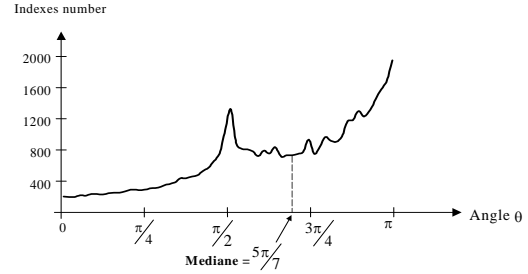


Figure 4: θ distribution

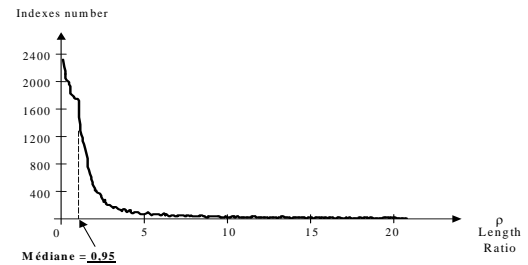


Figure 5. ρ distribution

It has been shown [4] that feature uniform distribution performs better indexing retrieval results. Therefore, we considered logarithmic function $\ln(\rho)$ (figure 6) which shows that values beyond the bound $[-4, +4]$ are not relevant.

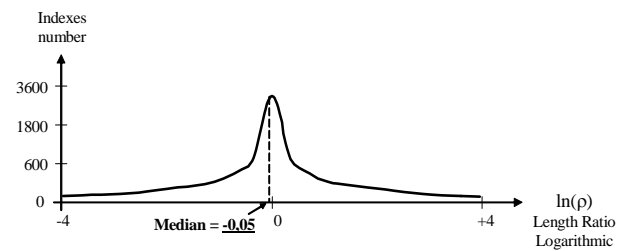


Figure 6 : $\ln(\rho)$ distribution.

2.4 Features Index composition

We proposed the following image features and index:

$\ln(\rho)$	θ	R	G	B	θ_0	l_1	l_2	x_0	y_0	No Imag
-------------	----------	---	---	---	------------	-------	-------	-------	-------	---------

index

For each image, a set of features are calculated, and the corresponding indexes are coded in a structure.

3 Object database

3.1 Features structure

We considered 3D polyhydic objects. For each object, several images are taken from different points of view (figure 7). For each image, adjacent intersecting segments and corresponding geometric quasi invariants are calculated. The corresponding set of features and indexes are added to the object database structure.

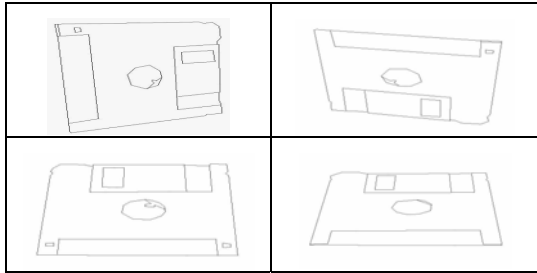


Figure 7. Object images.

Building the object database is the off line indexing process. The on line retrieval process good performance are much more of our concern. Therefore, we designed the database to fulfil the retrieval process needs: velocity and accuracy. To achieve these needs, we proposed two storage structures:

- One for the indexes : kept in the primary memory, which makes database access for retrieval process faster,
- One for image features: kept in a secondary memory.

On the other hand, for efficient query processing in large data sets, it is necessary to build an index structure that reduces the size of the retrieved set needed to answer a query. The general approach is to prune the search space and eliminate irrelevant data objects without accessing the corresponding features subspace. This had led to several index structures such as VQ tree [8], R*tree [9], X-tree [10]. In our case, image features lead to a multi dimensional index space. We've adapted the X-tree indexing structure to avoid irrelevant overlapping hyper cubes and empty leaves.

3.2 Indexes X-tree structure

When adding data to the database, X-tree leave split process creates leaves misdistribution. To create new leaves with balanced data distribution, a gravity center driven split process is used (figure 8).

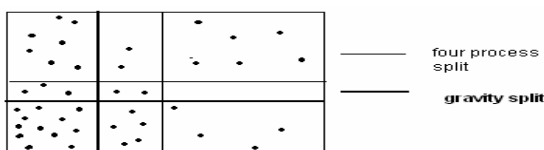


Figure 8. Split processes

3.3 Indexes similarities and VA-File approximation: similarity space

Due to image features quasi invariants properties, dense clusters in index space correspond to database objects. Finding dense clusters is a conventional nearest-neighbour approach that uses multidimensional access method. Unfortunately, while this method performs well for a low dimensionality, performance degrades as dimensionality increases.

To overcome this well known “dimensional curse”, we introduce vector approximation file descriptor [13] to analyze clusters density. The VA-file method is a geometric approximation that split in 2 each data space dimension d_i (this is coded on d_i bits). The index space is then split in 2^b hyper cubes ($b = \sum d_i$). Each index is coded by the interval number it lies in. To avoid poor or empty partitions, intervals with less than one index are ignored (figure 9 shows a 2D index space case).

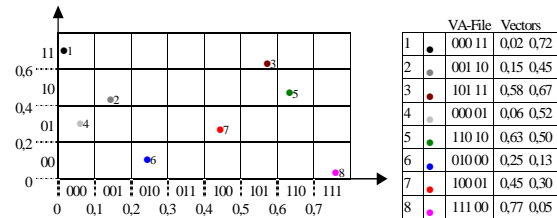


Figure 9. VA-file approximation

4 Retrieval process

It is a two step process. First, image features and indexes are extracted from the request object image, and then approximate in the similarity space. The second step is a match step: similarities of these indexes with neighbours are calculated and a vote eliminates the false matches.

4.1 Indexes similarity estimation

Similarities are calculated for each indexes neighbour. The VA-file answering request seeks then for the closest hyper cubes to the request. Euclidian distance is measured in between hyper cubes, and those for which the distance is bigger then a predefined ϵ value are rejected. This step is more like a filter that eliminates all non suitable hyper cubes.

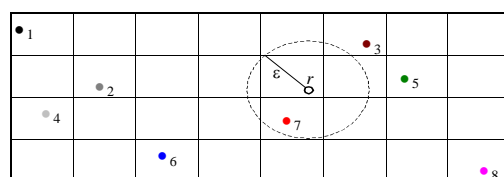


Figure 10. Request process

The result of these processes is a list of possible matches and corresponding similarities which forms clusters. After analyzing these clusters, we find the best request matches by a vote step process.

4.2 Vote step

The best match is the one with the higher density cluster. Our vote process is a three steps process:

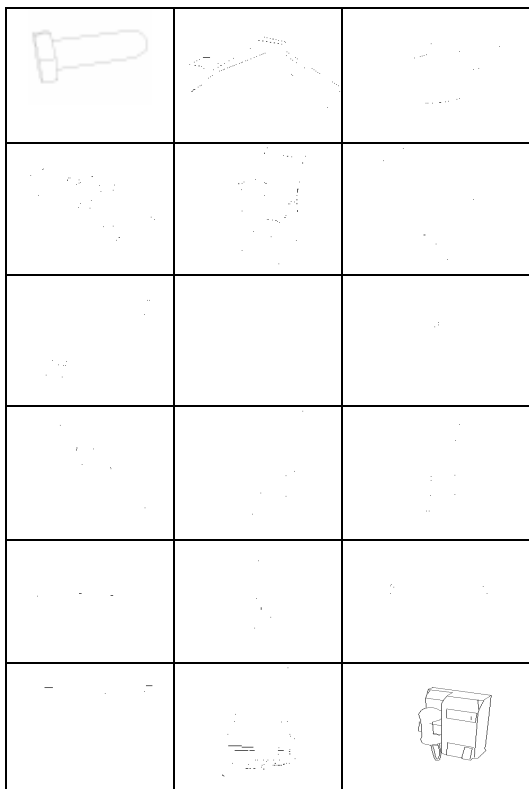
- Initialization: for each request image, an empty hypercube counters list, labeled with hypercube numbers, is created.
- For each similarity, we determine the interval number it lies in (i^{eme} dimension that contain its j^{eme} component). This gives us the hypercube number that contains these similarity parameters.
- If the hypercube number already appears in the initial list, its counter is incremented. Other wise, the hypercube number is added to the list and its counter set to 1.

When all the similarities have been treated, the matched indexes refer the more resembling objects in the database. The corresponding images are selected and sort out by the vote number.

5 Evaluation

5.1 Object Database

We considered 28 polyhydic objects seen under several point of view for each of them (average object rotation is 20^0). From the resulting 856 images we extract the geometric features and build the indexes.



5.2 The tree composition

During the object database building process, several tests have been conducted. We were specially concerned by the tree leaves composition.

5.2.1 Leaf fullness

Test have been conducted on hole, half and third index database (figure 11). The best fullness rate for leaves is achieved with 400 indexes.

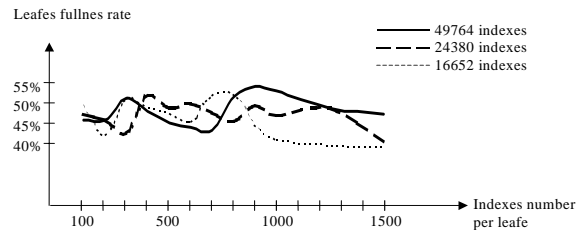


Figure 11. Leaf fullness

5.2.2 Leaf size versus request time

We tested a long request index (100 images indexes). Over 800 request indexes, the request time changes slightly (figure 12). Therefore, the maximum number of indexes stored in each tree leaves is 400 indexes.

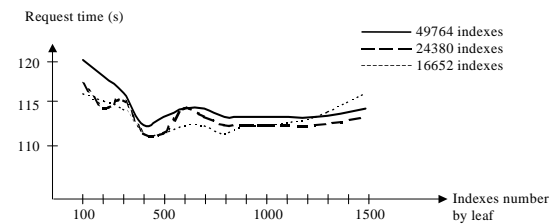


Figure 12. Time request

5.3 Indexes distance threshold

We evaluate the threshold used to compare indexes. With a set of 24 views of the same object, we analyzed identical geometric configurations and evaluate the difference between their quasi invariants features (figure 13). Half these values is the threshold:

$$(\ln(\rho), \theta) \leq (0,14, 11,2)$$

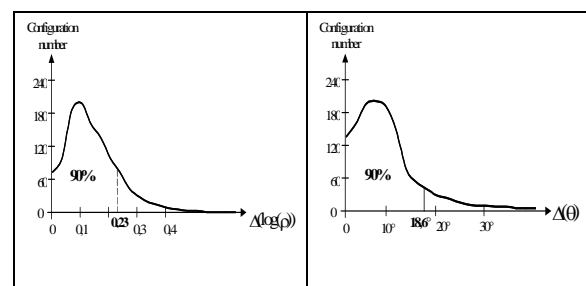


Figure 13. Similarities of quasi-invariants

5.4 VA-file Parameters

In the VA-file process, the similarities representation space is split into 2^b hyper cubes, and each dimension is divided in 2^{b_i} intervals, with the condition:

$$b = \sum_{i=1}^4 b_i$$

The VA-file parameters are the hyper cubes bounds and the intervals number b_i . They are defined in regard to the similarities variations. In our case, the similarities are homothety k , rotation α , and translation over X and Y axis.

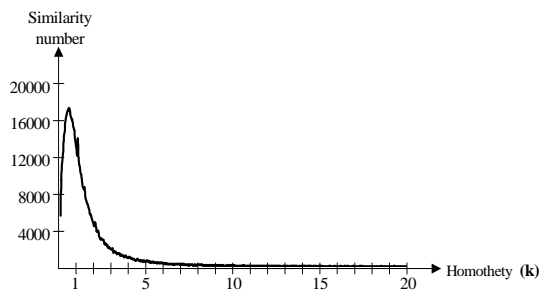


Figure 14. Homothétie k distribution

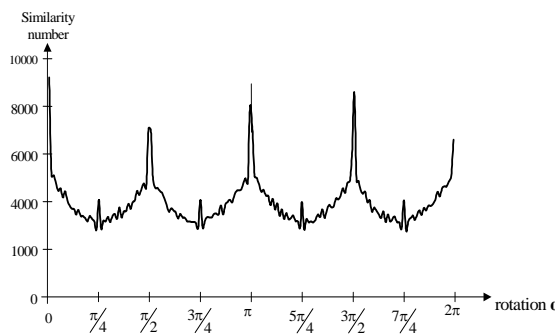


Figure 15. Rotation α distribution

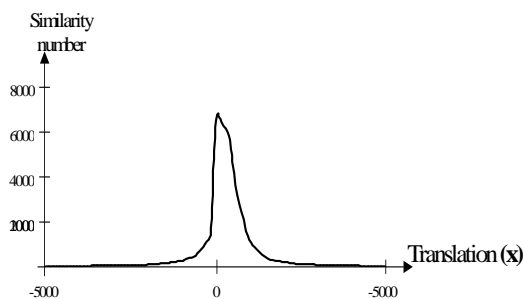


Figure 16. Translation X distribution

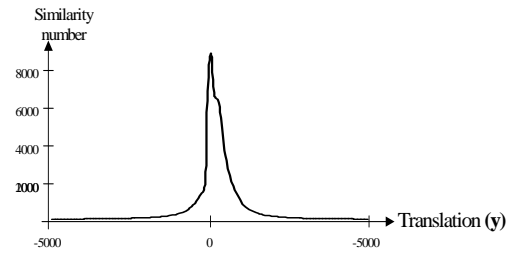


Figure 17. Translation Y distribution

The homothety distribution (figure 14) is a non uniform distribution. To ensure a uniform distribution, a logarithmic function is used (figure 18).

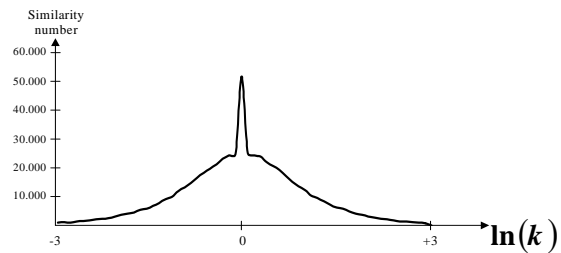


Figure 18. Homothety distribution $\ln(k)$

The interval numbers for each dimension is determined by the existing similarities (figure 14, 15, 16 and 18). We evaluate average similarity distance which leads us to these intervals numbers:

	Interval	gap	$b_i = \text{Interval} / \text{gap}$
k	$[-3, +3]$	0,13	$b_k = 6$
α	$[0^\circ, 360^\circ[$	4,32	$b_\alpha = 6$
X	$[-5000, +5000]$	49,83	$b_x = 8$
y	$[-5000, +5000]$	61,21	$b_y = 8$

5.5 Results

We show below a 3D object request image and the best matched images from the image database. The object (a disk) is well identified even if two different objects looked similar. This vote rate is due to an incomplete description of this object. The Quasi invariant features used to describe and identify objects from images didn't provide a unique description. We propose in a future work a new object description that insures uniqueness of the description.

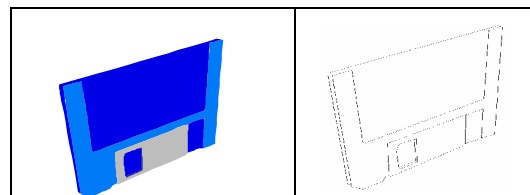
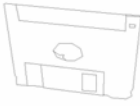









Figure 19. Object request Image

The request image has an average of 55 indexes. The response of the request from the geometric model

base gives an average of 241 similarities for each request index. The matching rate is 63%. In the fail case, the request image is matched with the right object with 58% vote rate and with wrong objects with 22% vote rate.

 Image 1 (Vote: 73)	 Image 2 (Vote: 12)
 Image 3 (Vote: 12)	 Image 4 (Vote: 11)
 Image 5 (Vote: 10)	 Image 6 (Vote: 9)
 Image 7 (Vote: 9)	 Image 8 (Vote: 8)

6 Conclusion

We proposed in this paper an efficient method for object indexing and retrieval from database. Our method for coding the indexes and their similarities improved the request time. The test with polyhydic objects images, taken with different points of view, shows good matching rates.

The use of geometric quasi invariants features as indexes make the use of images, regardless to the way they've been taken and without any information on the point's view possible (no calibration parameters are needed). Indexes can in the future be a combination of geometric and photometric features (reflectance). This combination will be used during the vote process to reject indexes which geometric similarities that leads to false matching.

7 References

- [1] A. R. Pope, D. G. Lowe, "Learning object recognition models from images", ICCV'93.
- [2] H. Murase, S. K. Nayar, "Visual learning and Recognition of 3D objects from appearance", Int. Journal of Computer Vision, 1995.

- [3] M. Daoudi, S. Matusiak, New Multiscale Planar Shape Invariant Representation under a General Affine Transformations. ", ICPR'2000, Barcelona, Sept. 2000, Vol.3, 794-797
- [4] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, "Content-based Image Retrieval at the end of the Early Years", IEEE PAMI, Vol. 22, No. 12, Dec. 2000.
- [5] S. MATUSIAK, "New Multiscale planar Shape invariant Representation under a General Affine Transformations", 1999.
- [6] Joseph L. Mondey & Andrew Zisserman, "Geometric invariance in computer vision", MIT press, 1992.
- [7] L. Amsaleg, P. Gros, S.-A. Berrani, "A Robust Technique to Recognise Objects in Images, and the DB Problems it Raises", Proc. of the Workshop on Multimedia Information Systems, Capri, Italie, November 2001
- [8] E. Tuncel, H. Ferhatosmanoglu, K. Rose, "VQ-index : An index structure for similarity searching in Multimedia Databases", ACM Multimedia, Juan Les Pins, France, December 2002.
- [9] N. Beckmann, H. Kriegel, R. Schneider, B. Seeger, "The R* tree: An efficient and Robust access method for points and rectangles", In Proc. ACM Sigmod Int. Conf. on Management of Data, May 23-25 1990.
- [10] S. Berchtold, D. Keim, H.P. Kriegel, "The X-Tree : An index structure for high dimensional data ", In Proceedings of the International Conference on Very Large Data Bases, Bombay, India, 1996.
- [11] M. J Fonseca, J. A. Jorge, "Towards Content Based Retrieval of Technical Drawings through High Dimensional Indexing", Computers and Graphics, 2003
- [12] C. Li, E. Chang, H. Garcia-Molina, G. Wiederhold, " Clustering Approach for approximate Similarity Search in High Dimensional spaces", IEEE Transactions on Knowledge and data engineering, 14(4):792 808, July August 2002.
- [13] R. Weber, H.J. Schek, S. Blott, "A quantitative analysis and performance study for similarity-search methods in high dimensional spaces", Proceedings of the 24th VLDB International Conference on Very Large Data Bases, New York, US, August 1998.

Detection of Cirrus Streak Utilizing Cloud Shape and Movement

H.Ikeda, R.Saegusa, S.Hashimoto

Department of Applied Physics, Waseda University, Japan.

Email: hiro@shalab.phys.waseda.ac.jp, ryos@ieee.org, shuji@waseda.jp

Abstract

For understanding atmospheric structure, it is important to detect cloud patterns in satellite images, since cloud pattern is closely related with atmospheric state. In this paper, we propose an image processing method to detect cirrus streaks of cloud pattern in satellite images utilizing a cloud movement. The proposed method is comprised of a detection step, an evaluation step and an unification step. In the detection step, some candidate streaks are detected in the area with a large variance of the brightness and a large velocity, which are regarded as a cirrus cloud. The detection is based on the Hough transform. In the evaluation step, every candidate streak is evaluated with its shape and movement. In the unification step, more than two close streaks are unified according to their positions and movements. Some experimental results are shown to verify the proposed method.

Keywords: cirrus streak, cloud shape, cloud movement, Hough transform, automatic interpretation

1 Introduction

In meteorology, weather forecasters analyze the atmospheric phenomenon based on observational data and their knowledge. They then extract useful information for weather forecast such as temperature changing and atmospheric pressure pattern. Cloud patterns visually shown in satellite images are closely related with the distributions of temperature and moisture, atmospheric flow and so on. Hence, it is important to detect cloud patterns to understand atmospheric structure. Some cloud patterns such as frontal areas, fogs [1] and contrail [2] came to be extracted automatically utilizing textual features such as brightness and its deviation. However, there exists some cloud patterns which can not be detected automatically only using the textural features. One of them is the cirrus streak which is important for locating the jet stream. Consequently, meteorological experts detect cirrus streaks manually at present.

It is known that cirrus streaks have some features in its texture, shape and movement. One of textural features is that the cirrus streak, which is an upper layer cloud, containing small streaks vertical to the cirrus streaks. Regarding features of the shape and movement, the cirrus streak is elongated and flows fast along the flow of the upper layer. Considering the cloud shape and movement are effective features to express the dynamics of the cloud, we propose a novel method to detect the cirrus streak utilizing the features of the cloud shape and movement in addition to textural features.

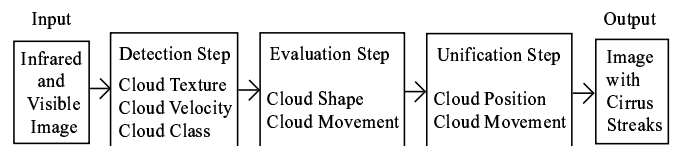


Figure 1: Overview of the proposed method.

In Section 2, we describe the detection of cirrus streaks based on the Hough transform. In Section 3, we show the experimental results with various satellite images to confirm the effectiveness of the proposed method. Finally, we give the conclusions and perspectives in Section 4.

2 Proposed Method

We firstly explain the overview of the proposed method. Figure 1 shows the overview. As an input we use infrared images and visible images taken from a meteorological satellite at different frequency bands. In general, visible images are used to see lower clouds and infrared images have advantage in understanding the dynamics of upper layer clouds. The brightness of the visible image corresponds to the thickness of the cloud, while the brightness of the infrared image corresponds to the temperature of the cloud, namely the top altitude of the cloud. As shown in Figure 2 and Figure 3, a cirrus streak surrounded by the dotted line in an infrared image is more distinct from the other clouds than that in a visible image. Hence, in the



Figure 2: Infrared image with the cirrus streak surrounded by the dotted line.

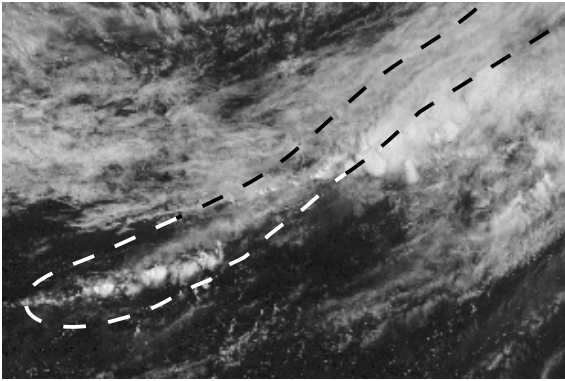


Figure 3: Visible image with the cirrus streak surrounded by the dotted line.

proposed method we mainly use infrared images, which are observable throughout the day and we use visible images only for the decision of the cloud class. Utilizing satellite images all the clouds are classified into a certain cloud class such as cirrus, cumulonimbus and so on [3][4].

The detection of cirrus streak is performed in three steps; the detection step, evaluation step and unification step. In the detection step, the objective pixels are selected according to the cloud class, cloud velocity and deviation of the brightness in the respective neighborhood. Then, candidate streaks are detected based on the Hough transform for the objective pixels. In the evaluation step, some candidate streaks are omitted by evaluating candidate streaks with their movement on the streak and their shape. In the unification step, conclusive cirrus streaks are determined by unifying close candidate streaks according to their positions and movements. Finally, an infrared image with the detected cirrus streaks is output. Details of the detection are described as follows.

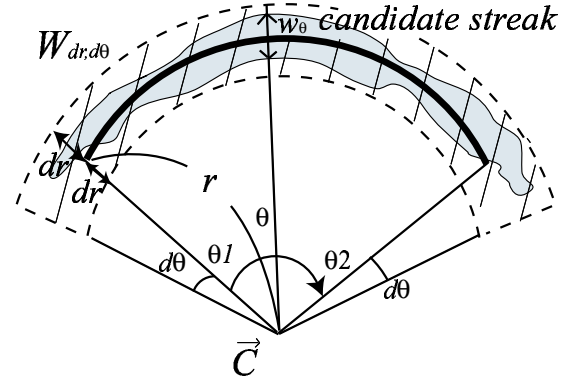


Figure 4: Parameter sets and $W_{dr, d\theta}$.

2.1 Detection of the candidate streak

In general, a cirrus streak is a straight or a slightly curved line. We therefore approximate a cirrus streak as an arc of a circle or its combination. The candidate of a cirrus streak is detected based on the Hough transform [5][6]. In the Hough transform, a figure which we want to detect in the image is described as parameters of the equation of circle. The reliable parameters are selected in the parameter space by voting. The Hough transform is known to be robust to the noise and discontinuities of the figure. By selecting appropriate parameters, multiple curves can be detected at one time. Three parameters are needed to represent a circle.

We define \vec{C} and $r (> 0)$ as the vector of a circular center and the radius of a circle, respectively. A circle is described as a parameter set (\vec{C}, r) shown in Figure 4. Pixels for voting are selected based on certain conditions as given in the procedure 1. The selected pixel in an image casts a vote to the corresponding grids in the parameter space. We then select parameter sets corresponding to the grids voted a lot. The arcs of the cirrus streak are obtained by selecting the appropriate areas of the circle. Cirrus streaks detection can be summarized as the following three procedures.

procedure 1: selection of voting pixels

Considering that a cirrus streak belongs to the cirrus class, flows fast and has a small wavy cloud lines which are almost perpendicular to the flow of the streak, we applied the Hough transform to the pixel which belongs to the cirrus class and has a large deviation of brightness and a large velocity. The calculated deviation is the standard deviation of the brightness of an infrared image in the neighborhood of 13×13 pixels. The cloud class of a pixel is determined by its brightness and deviation in a visible image and an infrared image. Details are

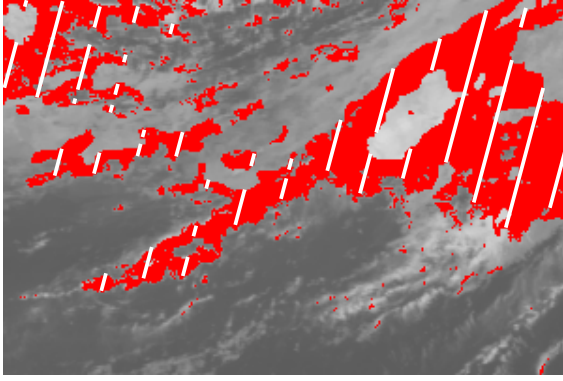


Figure 5: Infrared image including pixels for voting.

described in previous works [3][4]. The velocity of a cloud is obtained utilizing a pair of successive infrared images. The details are described in Section 2.2.

We define U as a set of pixels which belong to the cirrus and have a larger brightness deviation and velocity than a given threshold T_d and T_v , respectively. U are illustrated in Figure 4 and Figure 5 as the gray areas and the shaded areas in the infrared image, respectively. We performed the Hough transform to the pixels which belong to U .

procedure 2: decision of parameters

All pixels belonging to U are voted to a parameter space. Then, some parameter sets are selected in order of the votes. To detect appropriate parameter sets, we introduce the threshold T_h for the number of votes to select the remarkable parameter set in a parameter space. The threshold T_h is determined empirically. It should be small to detect the short curved line which is difficult to be detected by the Hough transform. Two close points in the parameter space are regarded as the same circle in the image. Therefore we select the parameter set which locates away from the already-selected parameter sets by 100 pixels for each parameter in a parameter space.

We assume that pixels which have pixels belonging to U in its neighborhood of 5×5 pixels are a part of a cirrus streak. Arcs comprised of only these pixels are selected from each circle as the candidate arcs, only when their length of the arc is beyond the threshold T_l . Let us define θ_1 and θ_2 ($\theta_1 \leq \theta_2$) as the angles at end-points of the arc. Then, a candidate streak is represented as a set of (\vec{C}, r) and (θ_1, θ_2) , as shown in Figure 4. An angle is measured in anticlockwise direction.

procedure 3: correction of the candidate streak

Each of candidate streaks obtained in the procedure 2 has to be corrected, since performing the Hough transform in the whole area which is much larger than the cirrus streaks, causes the failure detection. To exclude the influence, the Hough transform is performed in the neighborhood of the candidate streak, again. Let us define W_{d_r, d_θ} as the neighborhood area of the candidate streak which is illustrated in Figure 4 as the shaded area surrounded by the dotted line. The second Hough transform is applied to the area W_{d_r, d_θ} . Each parameter set is updated to that with the maximum votes. The candidate streak is obtained as well as the procedure 2.

2.2 Evaluation of the candidate streak

In the evaluation step, we use two features to classify the candidate streak as a cirrus streak or not. When the conditions for these features are satisfied, the candidate streak is assumed to be a cirrus streak. We define F_1 and F_2 as a feature of a cloud shape and a feature of a cloud movement, respectively. Each feature is described as follows.

feature 1: cloud shape

To evaluate the shape of candidate streaks, we utilize the average width of the candidate streak. Let us define w_θ as the width of the candidate streak at the angle θ , as illustrated in Figure 4. w_θ is comprised of consecutive pixels which have pixels belonging to U in their neighborhood of 3×3 pixels. F_1 is calculated by the following Equation (1).

$$F_1 = \frac{1}{N_1} \cdot \sum_{\theta_1 < \theta < \theta_2} w_\theta, \quad (1)$$

where N_1 is the number of pixels on the streak. As the shape of the candidate streak is more slender, F_1 gets smaller. A candidate streak with F_1 over the threshold T_1 is omitted. As the feature for the streak shape, the deviation of the width and the distribution of parameters in parameter space are assumed alternative. However, experimental result suggests that the average width is best to evaluate the streak shape.

feature 2: cloud movement

Each pixel of clouds has a movement vector as illustrated in Figure 6 To obtain the cloud movement we use the normalized cross-correlation method with variable template size adapted to the cloud state. To obtain the cloud velocity

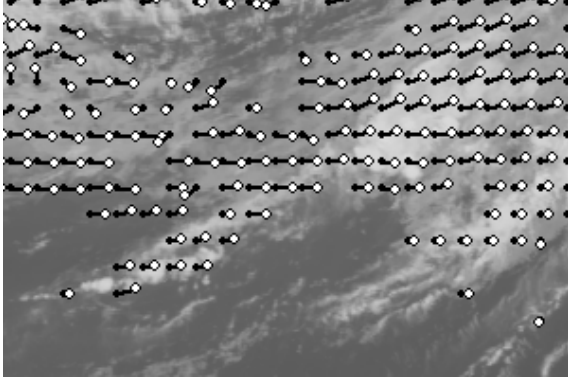


Figure 6: Movement vectors. (Black dots and white dots mean the origin and the end of each movement vector, respectively.)

from an image at a given time, we have to detect the corresponding cloud in the image of the next frame. In the normalized cross-correlation method [7], a template from the first image is matched to a remarkable area in the second image by shifting its position pixel by pixel. The pixel which has the maximal correlation value is assigned as the corresponding pixel. However, we often lose the correct corresponding pixel in the cloud area, since the variance of brightness on a cloud area is small for a temporal change. Utilizing not an area of the clouds but a boundary of cloud area, we can obtain a more accurate velocity, since the variance of the brightness in a boundary is relatively large. To include enough boundary to obtain the accurate velocity, the template size is adaptively enlarged until the number of pixels in the boundary exceeds a threshold T_b determined empirically. In the experiments, the initial size of template and T_b were set as 31×31 pixels and 100 pixels.

To consider whether the flow of the remarkable streak is fast and along the flow of the upper layer clouds, we define F_2 as the summation of an inner product of the tangential unit vector and the movement vector of the clouds on the remarkable streak. Let us define $\vec{v}_{s,\phi}$ and \vec{n}_θ as a movement vector of the cloud and a tangential unit vector of the arc as illustrated in Figure 7. F_2 is calculated by Equation (2).

$$F_2 = \frac{1}{N_2} \cdot \sum_{\theta_1 < \theta < \theta_2} \vec{v}_{s,\phi} \cdot \vec{n}_\theta, \quad (2)$$

where N_2 is the number of pixels on the streak, the direction where the candidate streak flows is determined as the direction with the larger F_2 . F_2 gets larger, when the velocity of clouds is faster and the vector is along the flow of the upper layer clouds. The arc with F_2 below the threshold T_2 is omitted.

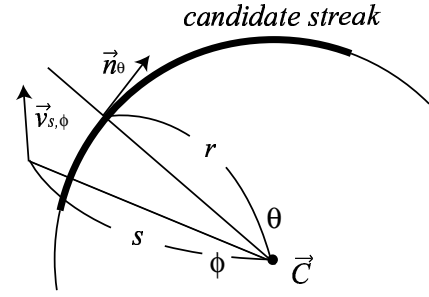


Figure 7: Tangential vector and movement vector.

2.3 Unification of the candidate streak

There exist some candidate streaks regarded as a cirrus streak in the image, although they disintegrate each other in the parameter space. To obtain a candidate streak fitting the cirrus streak, some streaks are to be unified when the both ends of one streak are in the region W_{d_r, d_θ} of the other streak. When unifying two streaks, the candidate streak with a larger F_2 remains as a candidate streak. Since the streak with large F_2 flows along the upper cloud, it can be regarded as a cirrus streak.

3 Experiments

Visible and infrared images of a GMS5 image database ¹ were used for the experiments. Sequential images acquired in July and August, 1999 at time 0:00UTC and 1:00UTC were chosen from the database. These images were taken every hour and the size is 500×500 pixels covering a large area of the Pacific Ocean around Japan (the spatial resolution is 5km/pixel). In the experiment, the thresholds were set as follows. (T_d : 8, T_v : 6, T_l : 100, T_h : 100, T_1 : 20, T_2 : 8). The parameter sets (d_r [pixel], d_θ [degree]) of W_{d_r, d_θ} were set as (40, 20). The thresholds concerning voting are not so sensitive parameters. Depending on them only the length of the detected streak can be changed. The thresholds concerning features are sensitive parameters, which effect on the determination of detected streaks.

To examine the effectiveness of the proposed method, we conducted some experiments for images with cirrus streaks and images without cirrus streaks. Two typical examples are given in Figures 8 - 13. Figure 8 and 11 are the input infrared images. Figure 9 and 12 are the results of the proposed method. The white lines, gray lines and black lines in these images represent detected streaks and omitted streaks in the unification step and evaluation step, respectively. Figure 10 and 13 depict the movements of the clouds.

¹<http://weather.is.kochi-u.ac.jp/>



Figure 8: Infrared images with two cirrus streaks surrounded by the solid line.

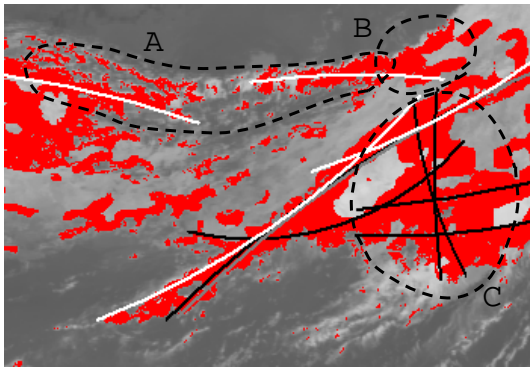


Figure 9: Result of the proposed method.

The calculated arcs fitted a slightly curved and straight cirrus streaks with an appropriate radius (See Figure 9 and 12). The cirrus streak A in Figure 9 shows that a cirrus streak which can not be approximated by a circle is expressed with a combination of two or more arcs and that a cirrus streak with slightly disconnected parts is detected as a combined streak. From these result, it seems that the proposed method can fit arcs to a cirrus streak robustly. However when clouds with a brindle texture exist in the neighborhood of a cirrus streak, the direction and the length of the detected streak were made different from the actual streak in consequence of these clouds (See the area B in Figure 9). A cirrus streak in the area where resultant cloud movements were wrong was not detected in the detection step (See the area B in Figure 12). As shown in the area C in Figure 9, the candidate streaks which were not a cirrus streak were omitted utilizing F_1 or F_2 . These results show that some candidate clouds are omitted by the cloud movement or cloud shape. However some actual cirrus streaks were incorrectly omitted by the condition of the cloud movement, since the movement vectors on the streak were not obtained properly. The cirrus streak A in Figure 12 shows that three candidate streaks are unified appropriately in the unification step.

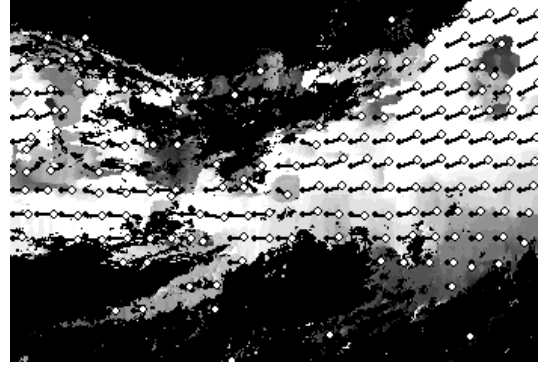


Figure 10: Movement vectors, of which the velocity gets larger as the brightness of its pixel gets larger.

Table 1: Experimental result of the decision about the detected streaks. (Each row of the table represents the number of detected streaks for each streak by the expert.)

	Cirrus streak	Not cirrus streak
Cirrus streak	12	5
Not cirrus streak	4	14

The experimental result of the decision about whether the detected streak is a cirrus streak or not is shown in Table 1. Each of the detected streaks was decided by the visual observation method compared to the expert's result. The total number of detected streaks was 35. The detection rate of actual cirrus streaks was 75%, and the error rate of not actual cirrus streaks was 26.3%. The streaks detected as a cirrus streak but not an actual cirrus streak tend to follow the upper layer clouds and to have a large textual change. They seem similar to a cirrus streak by appearances. From these result, it can be found that the proposed method is effective to detect cirrus streaks. However, small cirrus streaks comprised of a few pixels could not be detected in detection step.

4 Discussion and Conclusions

In this paper, we proposed a novel method for detecting cirrus streaks utilizing the cloud shape and movement. We demonstrated some experiments with various images to examine the effectiveness of the proposed method. A cirrus streak was detected based on the proposed method, when it had apparently features related to the cirrus streak. In the experimental result, there existed disconnected cirrus streaks and cirrus streaks crossing the other cirrus streaks. Hence, the process to connect such streaks should be added.

The experimental results showed that the cloud movement has a positive effect on classifying the

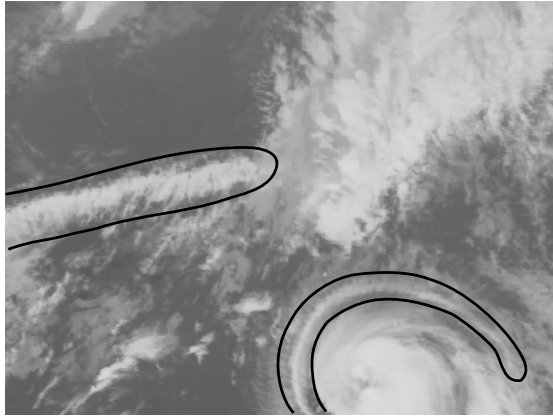


Figure 11: Infrared images with two cirrus streaks.

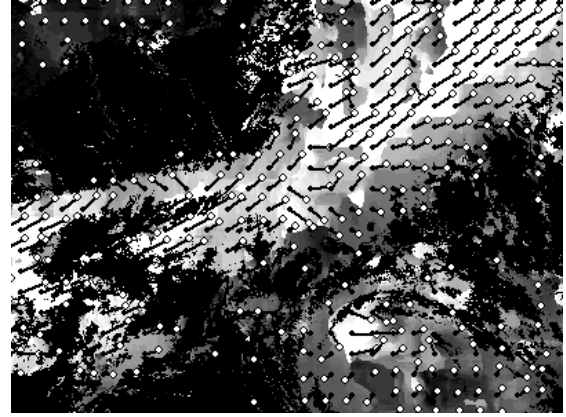


Figure 13: Movement vectors.

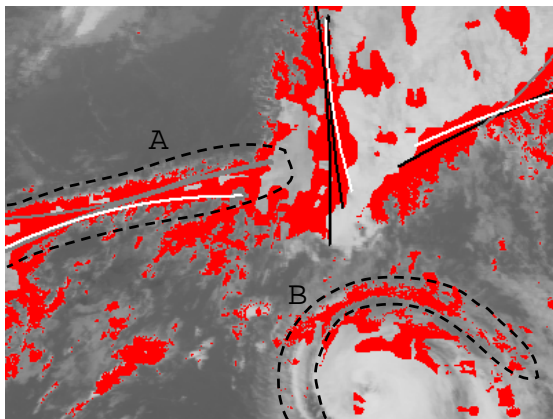


Figure 12: Result of the proposed method.

cirrus streak. Cirrus streaks which follow the flow of the upper layer clouds were correctly detected. However, cirrus streaks in the area where cloud movements can not be measured properly failed in the detection and the other types of cloud were incorrectly detected as the cirrus streaks. Since the detection accuracy depends on the precision of cloud movements, we are improving the method to obtain the accurate cloud movements based on its temporal and spatial consistency. In the future work, we will apply this method to the detection of the jet stream from water vapor images and streaks belonging to the other cloud classes such as Cumulonimbus and Cumulus.

Acknowledgements

This work was supported in part by the following funds: (i) "Establishment of Consolidated Research Institute for Advanced Science and Medical Care," Encouraging Development Strategic Research Centers Program, the Special Coordination Funds for Promoting Science and Technology, Ministry of Education, Culture, Sports, Science and Technology, Japan, (ii) "The innovative research

on symbiosis technologies for human and robots in the elderly dominated society," 21st Century Center of Excellence (COE) Program, Japan Society for the Promotion of Science, and (iii) the Grant-in-Aid for the WABOT-HOUSE Project by Gifu Prefecture.

References

- [1] Ellrod, G. P, "Advances in the detection and analysis of fog at night using GOES multispectral infrared imagery," *Wea. Forecasting*, 10, pp. 606-619, 1995
- [2] J. M. Weiss, S. Christopher, and R. M. Welch, "Automatic Contrail Detection and Segmentation," *IEEE Geoscience* 36(5), pp. 1609-1619, Sep 1998.
- [3] H. Ikeda, M. Matsumoto, S. Hashimoto, "Cloud classification of satellite image performed in two stages," In *Proceedings of the SPIE Electronic Imaging '2007 Conference*, Jan 28 - Feb 1, 2007. (Accepted for publication)
- [4] B. Tian, M. R. Azimi-Sadjadai, et al, "Temporal updating scheme for probabilistic neural network with application to satellite cloud classification," *IEEE Trans. Neural Networks*, vol. 11, pp. 903-920, July 2000.
- [5] R. O. Duda and P. E. Hart. "Use of the Hough transform to detect lines and curves in pictures," *Communications of the ACM*, 15, 1, pp 11-15, 1972.
- [6] J. Illingworth, J. Kittler, "A Survey of the Hough Transform," *CVGIP*, vol. 44, pp. 87-116, 1988.
- [7] Q. X. Wu, "A correlation-relaxation-labeling framework for computing optical flow - template matching from a new perspective," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(8) pp. 843-853, 1995.

Region-based MRF Model for Moving Object Segmentation

S. K. Hwang¹ and W. Y. Kim²

¹ HIT Lab NZ, University of Canterbury, New Zealand

² Division of Electrical and Computer Engineering, Hanyang University, Korea

Email: sunkyoo@vision.hanyang.ac.kr, wykim@hanyang.ac.kr

Abstract

This paper presents a new Markov Random Field (MRF) energy model for region-based moving object segmentation in video. The proposed MRF model is defined by a combination of normalized distances of regions, where mean colour, edge magnitude on the boundary, and motion of regions are considered. In addition, we introduce a weight term that adds region growing characteristics to the MRF model so that the accurate boundary of an object is acquired. The procedure for moving object segmentation is as follows. A watershed algorithm is used to partition every frame into a set of homogenous regions. As the watershed algorithm produces, in general, irregular shaped regions of various size, block-wise subdivision and small region merging is applied to improve the regularity of the regions. A region adjacency graph (RAG) is constructed and the motion vector of each region is estimated by a modified diamond search. Then, an MRF model with the proposed energy function labels the foreground or background of each region. The minimization of the energy function is carried out by a highest confidence first (HCF) algorithm. In the experiments, we confirmed that the proposed energy model has the region growing characteristics and segments the moving object accurately in various videos with or without camera motion.

Keywords: Moving object segmentation, MRF model, region-based, region growing, RAG

1 Introduction

Moving object segmentation in video is an essential process for analyzing video data. It is a base technology for object-based compression in MPEG-4 [1] and content-based retrieval in MPEG-7 [2]. Moving object segmentation has been studied in the form of Video Object Plane (VOP) extraction in MPEG-4 or motion layer extraction. Moving object segmentation is computationally expensive, but a region-based approach is a solution for reducing such computational complexity.

Tsaig and Averbuch proposed a framework for automatic segmentation of moving objects with a Markov Random Field (MRF) model [3]. They partitioned each frame into homogenous regions by using a watershed algorithm, and constructed a Region Adjacency Graph (RAG). The MRF model is defined on the RAG to acquire accurate segmentation results. Zeng and Gao followed that framework with a solution to occlusion problems [4]. They detected the occlusion region by a forward and backward motion validation scheme and obviated the potential misclassification of uncovered background regions. Zeng and Gao also proposed a hierarchical MRF model [5]. MRF models are generally used to assure spatial and

temporal consistency. However, they often fail to acquire the accurate boundary of the object, and thus a region growing technique is used to improve the segmentation results [4]. The region growing technique has also been used to obtain the accurate boundary of objects [6]. In that study, Kim et al. proposed a semi-automatic segmentation method, because full automatic moving object segmentation often fails when the motion information acquired is insufficient. The initial location of semantic objects was provided by user input. Kim et al. defined an uncertainty region on the object's boundary and found the accurate boundary by a bi-directional region growing technique.

In this paper, we present a new MRF energy model for region-based moving object segmentation in video. Basically, the energy model is defined by a combination of the normalized distances of the region's features such as mean colour, edge magnitude on the boundary, and motion. In addition, we introduce a weighting factor derived from the edge directional information of the regions. As the weighting factor imparts region growing characteristics to the energy model, the object's accurate boundary is gradually acquired. In the experiments, we show the performance of the proposed method with video segmentation results.

2 Region-Based Image Representation

To partition an image into a set of homogenous regions, in this study, we used a watershed algorithm that treats the input image as a topological surface and divides the image into homogenous regions [7]. The watershed algorithm has been widely used in region-based segmentation methods [3][4][5][6]. As the watershed algorithm is very sensitive to image noise, generally a noise reduction filter is applied as a pre-processing. In this study, an isotropic diffusion filter is used because it successfully removes the noise without destroying the topological structure of the image [8]. The watershed algorithm is applied to the gradient magnitude of the input colour image, which is computed in the YUV colour space. Let G_Y , G_U , and G_V denote the gradient magnitude of the three colour components Y , U , and V , respectively. Then, the gradient magnitude of a colour image, G_{col} , is computed by

$$G_{col} = \sqrt{G_Y^2 + G_U^2 + G_V^2}. \quad (1)$$

The watershed algorithm generally produces irregular shaped regions of various sizes, which fact increases the motion estimation error in the subsequent step [9]. Therefore, we subdivided the result of the watershed algorithm in a block-wise manner and merged the small regions to their neighbourhood. Figure 1 shows an example of the initial partitioning.

In order to estimate a motion vector of a region, we modified the diamond search (DS) algorithm [10]. The DS algorithm is one of the block matching methods for estimating a motion vector, and produces very fast and accurate results. In this study, we used a region instead of a block as the unit of template matching. Accordingly, every region had its own motion vector.

The region-based representation of an image is converted into a region adjacency graph (RAG). A vertex of the RAG contains the region's own features such as mean colour, shape, and its motion vector. A weight on an edge represents the similarity of two adjacent regions.

3 Segmentation by Region-Based MRF Model

Markov random field (MRF) model, a branch of probability theory, has been widely applied to the computer vision problem [11]. Since Geman and Geman used an MRF model in image restoration [12], many researchers have used MRFs in various areas of image processing including moving object segmentation [3][4][5].



(a) Result of watershed algorithm



(b) Result of block-wise sub-division of (a)

Figure 1: Examples of region-based image representation

Segmentation by MRF model can be obtained by labelling each region as either foreground (F) or background (B). By introducing a maximum a posteriori (MAP) solution, the segmentation results can be obtained by minimizing the posterior energy function U in

$$f^* = \arg \min_f U(f | d), \quad (2)$$

where $f = \{f_1, \dots, f_n | f_i = F \text{ or } B\}$ is a configuration of labels and d denotes the observations.

The segmentation procedures using the MRF model are depicted in a flow chart (Figure 2). Every frame in a video is transformed into a RAG, as described in section 2. The inputs to the MRF models are region-based representation of frame, motion information, and the segmentation result in the previous frame. Then, the MRF model provides the segmentation result of the current frame. At the first frame, a user input is used instead of the previous segmentation result.

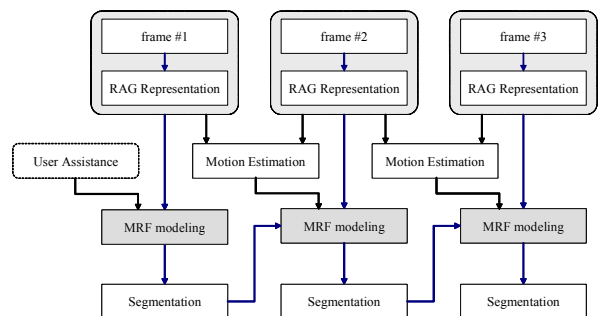


Figure 2: Block diagram of region-based moving object segmentation

3.1 Proposed Energy Model

The most important aspect of MRF modelling is the design of the energy function. The energy function should be defined so that its minimum corresponds to a good result. Large energy is interpreted as being unstable and likely to be changed. Modelling the energy function and finding a configuration that minimizes the energy function is the main procedure in MRF optimization.

The conditional posterior potential is generally written as

$$U(f_i | d_i, \hat{f}_{N_i}) = V(d_i | f_i) + \sum_{j \in N_i} V_c(f_i, f_j). \quad (3)$$

The first term on the right side in equation (3) is defined as

$$V(d_i | f_i) = \frac{1}{N_{R_i}} \sum_{(x,y) \in R_i} |f_i - S(x,y)|, \quad (4)$$

where N_{R_i} is the number of pixels in region R_i . Assume that f_i is 0 if region R_i is labelled as background, and 1 if foreground. $S(x,y)$ is a binary image and denotes the expected location of objects in the current frame. The potential $V(d_i | f_i)$ becomes zero if we assign an appropriate label to the region. $S(x,y)$ is given by a user at the first frame of the video and updated by the motion information and the segmentation result of the previous frame, as shown in figure 3. In the figure, n represents the frame number. This term guarantees temporal consistency.

In advance of the definition of the clique potential V_c , the following is assumed: If two labels of adjacent regions are the same, it is desirable that the regions have similar characteristics; that is, that the dissimilarity of those two adjacent regions with the same label be small. However, if the labels of adjacent regions are different, the dissimilarity would be large.

By using these properties, the clique potential is defined in two cases,

$$\begin{aligned} V_c(i, j | f_i = f_j) \\ = \alpha d_c(i, j) + \beta d_e(i, j) + \gamma d_m(i, j), \end{aligned} \quad (5)$$

$$\begin{aligned} V_c(i, j | f_i \neq f_j) \\ = \alpha [1 - d_c(i, j)] + \beta [1 - d_e(i, j)] + \gamma [1 - d_m(i, j)], \end{aligned} \quad (6)$$

where d_c is the colour distance, d_e is the edge distance, d_m is the motion distance of regions and α , β , and γ are the associated weight coefficients. The distance measures are normalized to the interval $[0, 1]$.

The colour distance is computed in the YUV colour space. If the mean colour of region R_i is defined in vector form, that is, $\mathbf{c}_i = (y_i, u_i, v_i)$, then the colour distance between two regions R_i and R_j is defined as

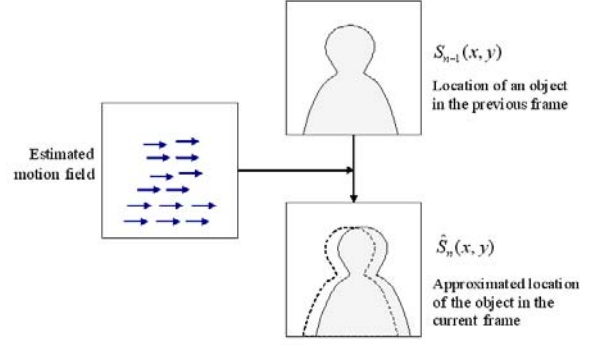


Figure 3: Update of $S_n(x, y)$

$$d_c(i, j) = \sqrt{(y_i - y_j)^2 + (u_i - u_j)^2 + (v_i - v_j)^2} / T_c, \quad (7)$$

where T_c is the normalization constant.

The edge distance of two adjacent regions is defined by the average value of the gradient magnitude on the common boundary:

$$d_e(i, j) = \frac{1}{N_{B_c}} \sum_{(x,y) \in B_c} |G_{col}(x, y)| / T_e, \quad (8)$$

where B_c is a set of pixels on the common boundary, N_{B_c} is the number of pixels in B_c , and T_e is the normalization constant.

Each vertex in the RAG contains a single motion vector of the corresponding region. To reflect human perceptual characteristics for motion similarity, we adopted the distance metric used in a previous study [13]. The motion distance measure is defined as

$$\begin{aligned} d_m(\mathbf{v}_i, \mathbf{v}_j) &= (\Delta^2 U_x + \Delta^2 U_y)^{1/2} \\ \Delta^2 U_x &= L_i \cos \theta_i - L_j \cos \theta_j, \\ \Delta^2 U_y &= L_i \sin \theta_i - L_j \sin \theta_j, \end{aligned} \quad (9)$$

where L_i , L_j , θ_i , and θ_j are calculated by

$$\begin{aligned} L(\mathbf{v}) &= \log(1 + k(v_x^2 + v_y^2)^{1/2}) \\ \theta(\mathbf{v}) &= \tan^{-1}(v_y / v_x) \end{aligned} \quad (10)$$

For the details of the motion distance, see [13].

3.2 Edge Directional Weight

Previous researchers have tried to use a region growing technique as a post process in order to acquire the exact boundary of the object [4][6]. If we assign a region growing characteristic to the energy model, the accurate boundary of objects can be detected more simply.

Let \mathbf{v}_g be a vector that indicates the normal direction of the boundary of objects, as shown in figure 4. The vector \mathbf{v}_g can be computed by the gradient vector field (GVF) method [14]. Figure 5 shows an example of GVF, where every vector points in the normal direction of the object's boundary. The vector \mathbf{v}_n is a normal contour vector of a region near the object. Then, the edge directional distance can be defined as

$$d_{ed}(\mathbf{v}_g, \mathbf{v}_n) = \frac{\mathbf{v}_g \cdot \mathbf{v}_n}{\|\mathbf{v}_g\| \|\mathbf{v}_n\|}. \quad (11)$$

If \mathbf{v}_g and \mathbf{v}_n point in the same direction, then d_{ed} is 1. Otherwise, d_{ed} is smaller than 1 and becomes -1 when the direction is completely opposite. Using d_{ed} , an energy function weight is defined by

$$w(d_{ed}) = \frac{1}{1 + \exp(-\lambda \cdot d_{ed})} + \frac{1}{2}. \quad (12)$$

The weight function has a value in the range of [0.5, 1.5]. Figure 6 shows the general form of the weight function with various values of λ .

By using the weight factor, equations (5) and (6) are modified as

$$V_c^*(i, j | f_i = f_j) = V_c(i, j | f_i = f_j) w(d_{ed}), \quad (13)$$

$$V_c^*(i, j | f_i \neq f_j) = V_c(i, j | f_i \neq f_j) w(d_{ed}). \quad (14)$$

The weight function emphasizes the relation of two adjacent regions with a high value of d_{ed} so that it imparts a region growing characteristic to the MRF model.

The minimization of the energy function is carried out by the highest confidence first (HCF) algorithm [15].

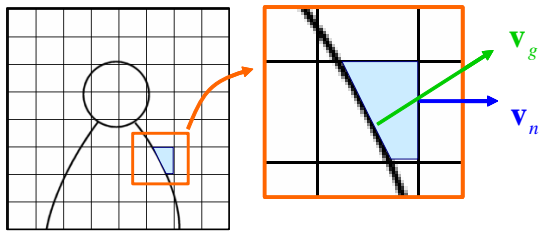


Figure 4: Edge directional information

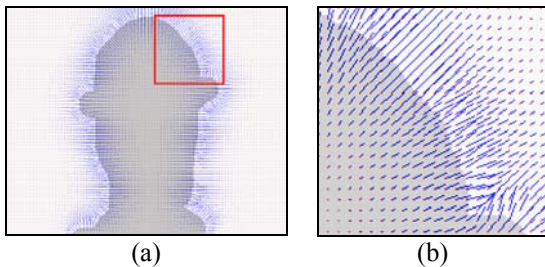


Figure 5: Samples of GVF (a) GVF for the human body silhouette (b) Enlarged GVF in the box in (a)

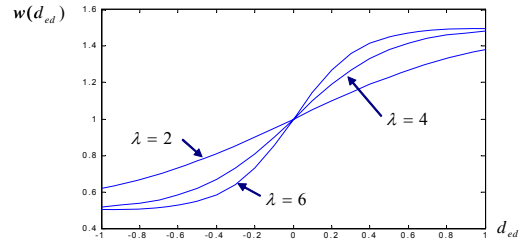


Figure 6: Function form of edge directional weight with various values of λ

4 Experiments

The proposed segmentation method was applied to video sequences with or without camera motion. The experiments were performed with a 3 GHz Pentium IV PC with 1 GBbyte RAM, and were implemented with Microsoft visual C++ 2005. The weight coefficients in equations (5) and (6) are set as $\alpha=0.7$, $\beta=1.0$, and $\gamma=0.5$. With the current implementation without any effort of optimization, the execution time for processing one frame size of 352×288 was about 1~2 sec.

Figure 7 shows the segmentation results from the 'mother & daughter' sequence. This video has no camera motion, and the movement of the mother and daughter is very small. At the first frame, the initial location of the object was revealed by user input, and is displayed in figure 7(a) as yellow. Figures 7(b)~(d) show the segmentation results without the edge directional weight. The results fail to segment the accurate boundary of the objects. However, these undesirable segmentation results are gradually refined when the edge directional weighting factor is used, as shown in figures 7(e)~(g).

Figure 8 show the segmentation results for a movie clip. In this video, a man moves from left to right and the camera follows the man, so the background is changing rapidly. The left columns show the original frames of video, and the right columns are the segmentation results for the selected frames. We confirmed that the proposed method extracted the boundary of the moving object successfully even though the video contained large camera motion.

5 Conclusions

In this paper, we presented a new MRF model that has region growing characteristics for region-based moving object segmentation. From the region growing characteristics, the accurate boundary of objects is gradually acquired. In experiments, we confirmed that the proposed method segments the moving object in videos with or without camera motion.

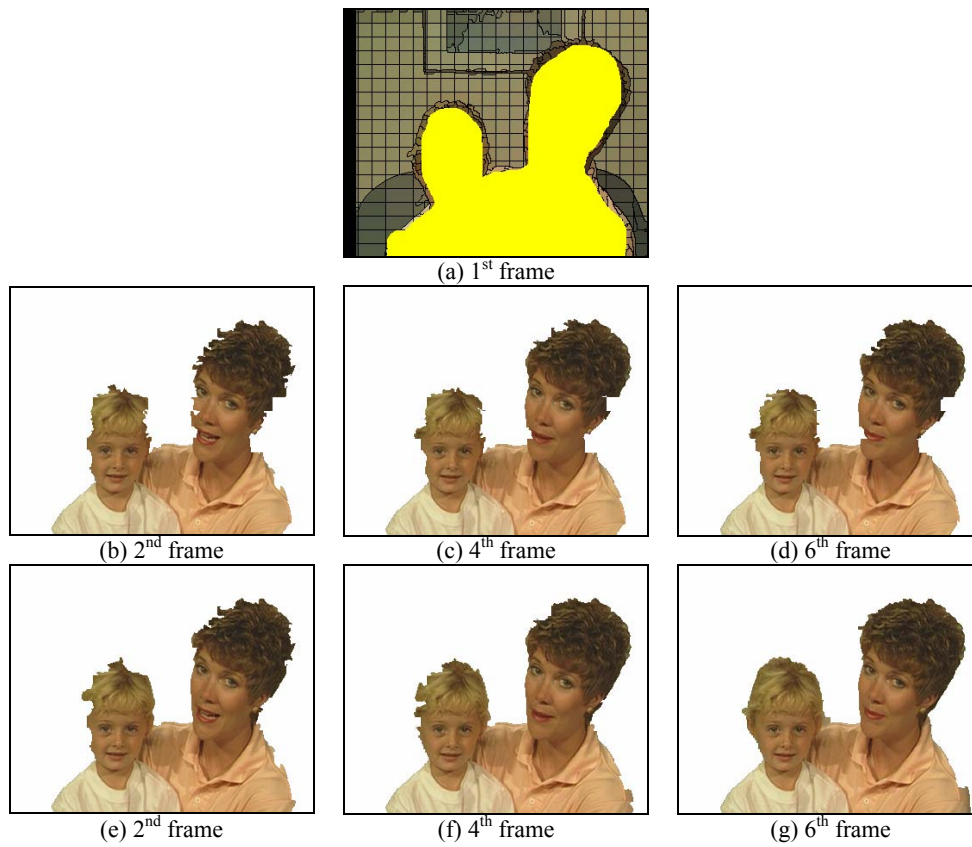


Figure 7: Segmentation results in a video without camera motion

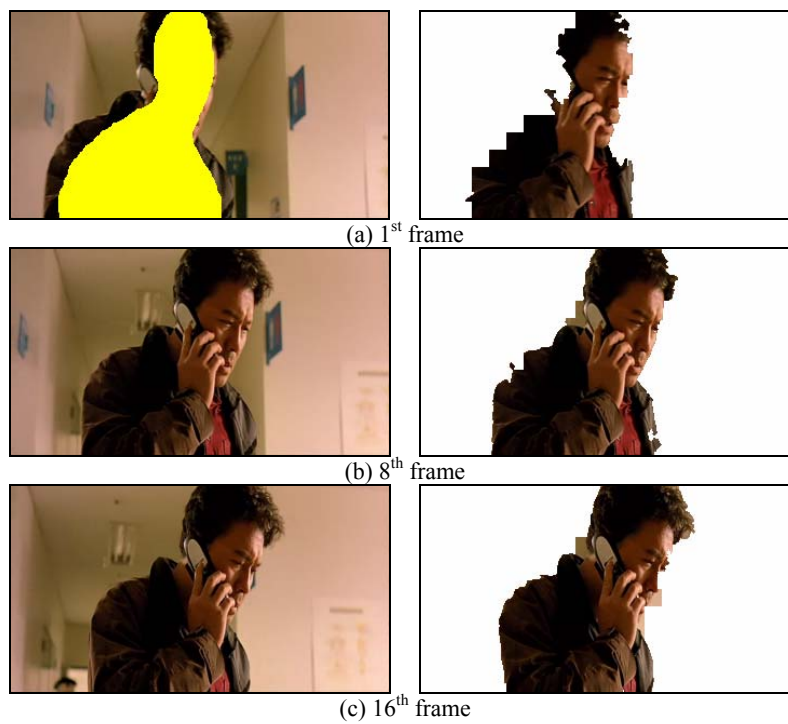


Figure 8: Segmentation results in a video with camera motion

6 References

- [1] MPEG-4 Video Verification Model Version 15.0, ISO/IEC JTC1/SC29/WG11 N3093, 1999.
- [2] M. Bober, "MPEG-7 Visual Shape Descriptors," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 11, no. 6, pp. 716-719, June 2001.
- [3] Y. Tsaig and A. Averbuch, "Automatic Segmentation of Moving Objects in Video Sequences: A Region Labeling Approach," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 12, no. 7, pp. 597-612, Jul. 2002.
- [4] W. Zeng and W. Gao, "Unsupervised Segmentation of Moving Object by Region-based MRF Model and Occlusion Detection," *Int. Conf. on Information, Communications & Signal Processing - Pacific-Rim Conf. On Multimedia (ICICS-PCM2003)*, Dec.15-18, 2003.
- [5] W. Zeng and W. Gao, "Accurate Moving Object Segmentation by a Hierarchical Region Labeling Approach," *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, vol. 3, pp. III 637-640, May 2004.
- [6] Y.-R. Kim, J.-H. Kim, Y. Kim, and S.-J. Ko, "Semiautomatic Segmentation Using Spatio-Temporal Gradual Region Merging for MPEG-4," *IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences*, vol. E86-A, no. 10, pp. 2526-2534, Oct. 2003.
- [7] P. De Smet and D. De Vleeschauer, "Performance and Scalability of a Highly Optimized Rainfalling Watershed Algorithm," *Proc. Int. Conf. on Image Science, Systems and technology*, pp. 266-273, July 1998.
- [8] P. Perona and J. Malik, "Scale Space and Edge Detection Using Anisotropic Diffusion," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629-639, July 1990.
- [9] R. Cucchiara, A. Prati, and R. Vezzani, "Object Segmentation in Videos from Moving Camera with MRFs on Color and Motion Features," *Proc. of Computer Vision and Pattern Recognition*, vol. 1, pp. I-405-I-410, June 2003.
- [10] S. Zhu and K.-K. Ma, "A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation," *IEEE Trans. Image Processing*, vol. 9, no. 2, pp.287-290, Feb. 2000.
- [11] S. Z. Li, *Markov Random Field Modeling in Image Analysis*, Springer-Verlag, 2001.
- [12] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.6, No. 6, pp. 721-741, Nov. 1984.
- [13] T. Yoshida, "Distance Metric for Motion Vector Histograms Based on Human Perceptual Characteristics," *Proc. International Conference on Image Processing*, vol. 1, pp. I-904-907, Sept. 2002.
- [14] C. Xu and J. L. Prince "Snakes, Shapes, and Gradient Vector Flow," *IEEE Trans. Image Processing*, vol. 7, no. 3, pp. 359-369, March 1998.
- [15] P. B. Chou and C. M. Brown, "The Theory and Practice of Bayesian Image Labeling," *International Journal of Computer Vision*, vol. 4, pp. 185-210, 1990.

Structured Combination of Particle Filter and Kernel Mean Shift Tracking

A. Naeem¹, S. Mills², and T. Pridmore¹

¹School of Computer Science & IT, The University of Nottingham, Jubilee Campus, Wollaton Road, Nottingham NG8 1BB, United Kingdom.

²Geospatial Research Centre (NZ) Ltd, Private Bag 4800, Christchurch 8140, New Zealand

Email: azn@cs.nott.ac.uk, steven.mills@grcnz.com, tpp@cs.nott.ac.uk

Abstract

Particle filters are a powerful and widely used visual tracking technology. Their strength lies in their ability to represent multi-modal probability distributions that capture and maintain multiple hypotheses about target properties. A potential weakness, however, is that the particle set can become diffused, dispersing across the image plane rather than clustering around the target. A number of solutions to this problem have been proposed, including the use of the more recently developed Kernel Mean Shift tracker to guide particles towards a local mode. While this hybrid Condensation/Mean Shift tracker is effective, in most cases the Condensation component is an unnecessary overhead: Kernel Mean Shift is a competent tracker that only needs the particle filter to deal with more ambiguous situations in which errors might be made. We therefore propose an alternative hybrid approach in which Kernel Mean Shift is the dominant tracking technology, with a small number of particles being generated, in a structured fashion, to explore further and so resist errors when confidence in the Mean Shift algorithm is low. The proposed algorithm, which we term the Structured Octal Kernel (SOK) filter, has been implemented and is compared with Condensation, Kernel Mean Shift and Hybrid trackers. The SOK filter provides the most robust results, with comparable accuracy, at the lowest computational cost.

Keywords: Tracking, particle filter, Mean Shift, kernel, hybrid.

1 Background and Motivation

Visual tracking has received much attention in recent years, with particle filtering [1] being one of the most successful and widely adopted approaches. The strength of the particle filter lies in its use of a set of discrete particles to represent multi-modal probability distributions that capture and maintain multiple hypotheses about target properties. Particle filtering is an iterative process in which particles are repeatedly selected, projected forwards using a motion model, dispersed by an additive random component, and evaluated against the image data.

Many particle filter-based trackers have been developed since Blake and Isard first introduced the Condensation algorithm [1]. The Auxiliary Particle Filter [2] selects particles in a more intelligent manner, making them concentrate around the true target and yielding better results. The approximation to the posterior is smoothed in the Regularized Particle Filter [3], while ICondensation [4] uses importance sampling to combine high and low-level information within Condensation. A survey of commonly used particle filters can be found in [5]

A potential weakness of the particle filter, however, is that the particle set can become too diffuse, spreading across the image plane rather than clustering around the target. When this happens particles tend to migrate towards local maxima in their evaluation function, becoming caught on clutter and losing track of the true target. A number of solutions to the problem have been proposed. The Annealed Particle Filter [6] uses annealing to smooth out the evaluation function, making the global maximum clearer and reducing the chance of particles becoming caught on local clutter. The Kernel Particle Filter [7] applies the Mean Shift hill climbing algorithm to the particle set to pull the centre of the particle distribution towards the target centre. The Kernel Particle Filter can be effective, but clusters weighted particles without further reference to the image data, assuming them to sample a unimodal distribution. This may not be the case.

Recently, Maggio and Cavallaro. [8] used the kernel Mean Shift tracking algorithm [9] to move particles towards local maxima on each iteration of Condensation [1]. Kernel Mean Shift tracking is a hill climbing approach which first computes the likelihood of each pixel in a circular search space

around the prior target centre being the next target centre, then moves the previous centre towards the maximum likelihood solution. The object model and candidate model both comprise probability density functions (pdfs) approximated by 2-D normalised histograms over the RGB colour space. The two dimensions are the ratios red/blue and green/blue. A kernel mask is used to give a higher weighting to pixels nearer the centre of the circular search region; making the algorithm more robust to target localisation errors and partial occlusions. Kernel Mean Shift tracking is an iterative process which continues until the Bhattacharya distance between the target pdf and the candidate pdf is either zero or a minimum value [9].

Kernel Mean Shift provides efficient and effective tracking as long as the target object does not move further than its own diameter or leave the search area between frames, a number of variations on the theme have been described. Yang et al [10] replace the Epanechnikov kernel used in the original formulation [9] with a Gaussian, while Leung and Gong [11] improve the efficiency of the method by computing the pdfs and Bhattacharya distance over only a small sample of the pixels in the search region. The search area is first segmented to identify foreground pixels, to which a uniform random sampling is applied.

Maggio and Cavallaro's [8] hybrid tracker combines Condensation with Mean Shift tracking to provide a system in which particles are alternately diffused by Condensation and clustered by Mean Shift. Multiple hypotheses are maintained by projecting a number of particles randomly around the prior position, but then hill climb towards the best target centre.

The hybrid tracker shows performance advantages over both Condensation and Mean Shift tracking, but also has some drawbacks. As the particles are randomly projected we need a good number to cover a given search space. Running N Mean Shift trackers, where N is the number of particles in the system, also makes the system computationally expensive. Furthermore, many of the particles coalesce during the Mean Shift phase, moving to the same hypothesis and making the representation redundant.

Mean Shift is a competent tracker and in many situations can maintain tracking without the multiple hypotheses represented by the particle set. While valuable in areas of high ambiguity, in most cases the Condensation component of the hybrid tracker is an unnecessary overhead. These observations lead us to propose an alternative hybrid approach in which Kernel Mean Shift is the dominant technology, with a small number of particles being generated, in a structured fashion, to explore further when confidence in the Mean Shift algorithm becomes low.

The proposed algorithm, which we term the Structured Octal Kernel (SOK) filter, is described in Section 2. The SOK filter has been implemented and

is compared with Condensation, Kernel Mean Shift and Hybrid trackers in Sections 3 and 4. The approach is discussed in Section 5 and conclusions are drawn in Section 6.

2 The Structured Octal Kernel Filter

The Structured Octal Kernel (SOK) filter is a kernel Mean Shift tracker augmented by a backup strategy triggered when confidence in the current location estimate is low. Confidence at time t is given by

$$C_t = (1.0 - bhata(t))$$

where $bhata(t)$ is the Bhattacharya distance between object model and image data at time t .

A user-defined threshold, T , is applied to C at each time step. If C_t is below threshold a set of eight independent kernel Mean Shift trackers are spawned, each with the same object model as the original but at locations designed to cover a search area around the current position estimate (Figure 1). When these additional trackers have also each converged, nine estimates of target location are available, each with an associated confidence level. The estimate with the highest confidence is selected and the process continues. This mirrors the hybrid tracker of [8]; the algorithm effectively generates eight evenly spaced particles when confidence in the Mean Shift is low.

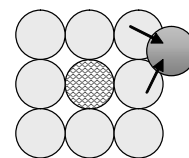


Figure 1. The SOK particle distribution. A hatched circle shows the primary KMS tracker, light circles the secondary "particles", a dark circle the target.

The SOK algorithm is therefore:

SOK Algorithm

1. Pick a target area (centre)
2. Compute a Normalized 2-D Histogram for area to get the target.
3. Loop {
 - Get a frame.
 - From the current centre compute the candidate normalized 2-D Histogram.
 - Compute Bhattacharya distance between target and candidate
 - Loop Till Bhattacharya distance Becomes constant
 1. Hill climb towards the maxima
 2. Compute candidate histogram again
 3. Compute Bhattacharya distance again
 - If Confidence < threshold value T
 1. Using current centre and radius place eight search areas in a structured manner as shown in the figure.
 2. Hill climb each particle towards the nearest maxima.
 3. Choose the one with the lowest value of Bhattacharya distance
 4. The chosen particle is the new target centre.

Normalisation ensures that the bin entries sum to 1, providing some robustness to movement along the line of sight. Note that r and T remain fixed throughout and that $bhata(i,j)$, and so C , varies between 0 and 1, easing selection of T , which is chosen empirically.

The SOK algorithm combines particle filtering with the kernel Mean Shift algorithm in a simple, but effective manner. Recognising the strength of the kernel Mean Shift algorithm in many situations it uses a single such tracker when confidence in the target location is sufficiently high. In areas of low confidence a burst of particles (cf. [8]) is emitted, allowing the tracker to search more widely. In the initial design the intention was to distribute these particles randomly. As there is no motion model in the Mean Shift tracker, however, and no prior distribution available to drive particle location, only a simple random distribution about the current location (e.g. uniform or Gaussian) was possible.

Noting the ability of the kernel Mean Shift tracker to climb to a local maximum if and only if the tracking window overlaps the target object, we adopt the simple particle distribution of Figure 1. This uses a small, fixed number of particles to cover a regular search area around the current hypothesis. To be beyond this search area the object would have to move more than twice its own radius between frames; which is unlikely. If high velocity motion is expected the particle set can be extended to create a larger search area, though in such circumstances kernel Mean Shift may not be the best approach and an explicit motion model may be required.

3 Experimental Evaluation

3.1. Algorithms

The proposed tracker has been experimentally compared with three existing algorithms. Here we briefly review the methods involved and describe their implementations. The image sequences used are presented and discussed in section 3.1.2.

3.1.1 Kernel Mean Shift Tracker

The kernel Mean Shift tracker [9] hill climbs from the previous location estimate toward a local minimum in the Bhattacharya distance between normalised, kernel weighted colour histograms representing the object model and local image data. We use a linear kernel having maximum weight at the centre and zero weight at boundaries and beyond. The object model and candidate model are 256 x 256 bin histograms recording red/blue against green/blue. This provides some robustness to changes in illumination. The histogram is normalised so the bin values sum to 1.

The Bhattacharya distance between model and candidate target is:

$$bhata() = \sqrt{1 - \sum_i \sum_j \sqrt{p(i,j) \times d(i,j)}}$$

where M is the size of each dimension of the histogram (256), and p and d are the object and the candidate models respectively. Note that the object model is computed only once. The candidate model is calculated in each frame from the position of the object in the previous frame.

The iterative Mean Shift operation is as follows:

$$x = \frac{\sum_i \sum_j \sqrt{\frac{p(i,j)}{d(i,j)}} \times i}{wt}$$

$$y = \frac{\sum_i \sum_j \sqrt{\frac{p(i,j)}{d(i,j)}} \times j}{wt}$$

where

$$wt = \sum_i \sum_j \sqrt{\frac{p(i,j)}{d(i,j)}}$$

where x and y are the coordinates of the next estimate of the position of the centre of the object, and M is again the resolution of the histograms modelling object p and candidate d .

3.1.2 Condensation

The particle filter used in the experiments conducted here is a straightforward implementation of Isard and Blake's [1] Condensation. The object and candidate models are exactly the same as those employed in the kernel Mean Shift filter, with Bhattacharya distance between them computed in the measurement phase. A simple motion model – constant velocity – is used throughout and, unless otherwise stated, all experiments use 100 particles.

3.1.3 Hybrid Condensation/Kernel Mean Shift Tracker

This again is a straightforward implementation of an existing technique – the hybrid tracker of Maggio and Cavallaro [8]. The Condensation algorithm outlined in section 3.1.2 provides a harness into which the Kernel Mean Shift tracker outlined in section 3.1.1 is slotted. At each time step 100 (unless stated otherwise) particles are evaluated by computing the Bhattacharya distance between the object and their candidate model. A further 100 particles are then selected with probability proportional to their measurement value and projected into the next image by a constant velocity motion model. A kernel Mean Shift tracker is initialised at each particle location and run until its associated Bhattacharya distance becomes zero or constant. The process is then repeated. This disperses the particle set in the Condensation phase, then draws it together in the Mean Shift phase (Figure 2).

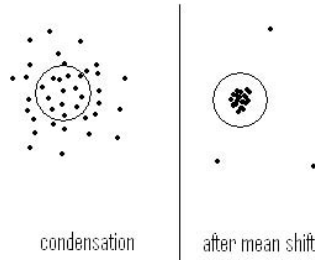


Figure 2. Mean Shift tracking alternately disperses and coalesces particles in the hybrid filter of [8].

3.2. Image Sequences and Evaluation Criteria

The four trackers described above have been evaluated and compared using a variety of real and artificial image sequences:

- Artificial sequences showing a multicoloured circular target moving across a white background allow the trackers' positional estimates to be compared to ground truth in the presence of controlled amounts of noise:
- To examine robustness to background clutter a hand-held ball is moved in front of a complex environment and viewed by a fixed camera. The sequence comprises 220 384 x 288 pixel frames, and is available from [12].
- To examine robustness to unpredictable motion, a hand-held camera is used to capture a 420 frame sequence of a child at play. Each frame is 720 x 576 pixels.
- To provide a quantitative comparison of the robustness of the four algorithms, McNemar's is applied to a set of 30 experiments on a variety of image sequences. All the sequences used here can be obtained from [13].

McNemar's statistic is a form of chi-square test for matched paired data. Consider the following 2×2 table of results for two algorithms:

	Algorithm A Failed	Algorithm A Succeeded
Algorithm B Failed	N_{ff}	N_{sf}
Algorithm B Succeeded	N_{fs}	N_{ss}

Table 1. Terminology used in McNemar's test

McNemar's statistic is then

$$x^2 = \frac{(|N_{sf} - N_{fs}| - 1)^2}{N_{sf} + N_{fs}}$$

where the -1 is a continuity correction. The central limit theorem states that if the sample size is moderately large and the sampling fraction is small to moderate, then the distribution is approximately

Normal. In such a case, the Z score (standard score) is obtained from (1) as:

$$z = \frac{(|N_{sf} - N_{fs}| - 1)}{\sqrt{N_{sf} + N_{fs}}}$$

Z value	Degree of confidence Two-Tailed prediction	Degree of confidence One-Tailed prediction
1.645	90%	95%
1.960	95%	97.5%
2.326	98%	99%
2.576	99%	99.5%

Table 2. Confidence limits associated with z value.

If the two algorithms give similar results then Z will tend to zero. As their results diverge, Z increases. Confidence limits can be associated with the Z value (Table 2). Two-tailed and one-tailed predictions are chosen according to the hypothesis: when testing if two algorithms differ, a two-tailed test should be used; when determining whether one algorithm is better than another, a one-tailed test is needed [14].

4 Results

Figure 3 shows the result of applying the four trackers to an artificial sequence in which a multicoloured target followed the path shown in Figure 4. This path comprises a number of straight sections corrupted by high levels ($\sigma = 10$ pixels) of Gaussian noise. Absolute error (in pixels) is plotted against frame number.

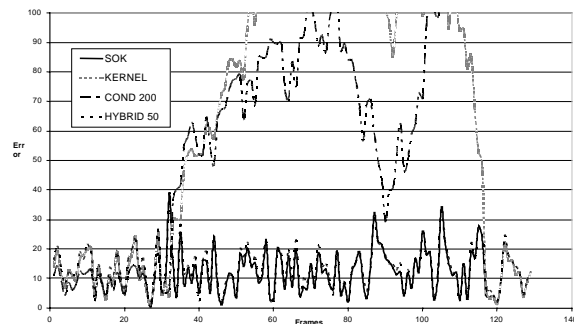


Figure 3. Absolute error (pixels) in four algorithms' tracking of a noisy (Gaussian, $\sigma = 10$) synthetic sequence showing a multicoloured target.

Kernel Mean Shift and Condensation both fail after the first sudden change in trajectory, while Maggio and Cavallaro's [8] Hybrid and the SOK filter track successfully. Note however, that the SOK filter used only one or eight particles, depending on tracking confidence, while at least 50 particles were needed to gain the same level of performance from the Hybrid.

McNemar's test [14] was applied to a set of 30 assorted image sequences [12] to provide quantitative comparison of the robustness of SOK with the Condensation, Mean Shift and Hybrid trackers. To apply McNemar, a definition of success and failure is

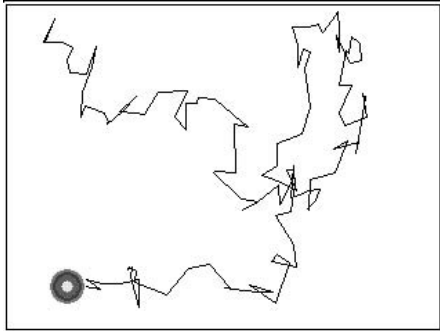


Figure 4. The multicoloured target moved over a white background to construct artificial test data.

required. Focussing on robustness, rather than accuracy of tracking, we define consider algorithm A to have succeeded and algorithm B to have failed if algorithm A maintains tracking for a greater proportion of a given image sequence, from the same starting parameters. In effect we define success to be tracking as long as the best of the two trackers. The results of this exercise are presented in Table 3, and show:

- 97.5% confidence that SOK is more robust than Mean Shift.
- 96% confidence that SOK is more robust than the Hybrid filter.
- 98% confidence that SOK is more robust than Condensation

	SOK vs. Mean Shift	SOK vs. Hybrid	SOK vs. Condensation
Z	2.041	1.809	2.219
Confidence	97.5%	96%	98%

Table 3. McNemar's comparison of SOK with Mean Shift, Condensation and Hybrid trackers over the image sequences available from [12]

Figure 5 shows selected frames from the four algorithms' tracking of a quickly moving, hand-held ball. Condensation fails after frame 35, when the particles diffuse towards different false local extrema. Kernel Mean Shift hovers around a confined area and loses the ball as soon as it moves quickly. Though it recaptures the ball later, when it passes under the Mean Shift window, this is not a robust effect. The Hybrid filter tracks quite well, but slips away a couple of times around frame 40. SOK tracks very well, using its structured backup when the ball slips away, e.g. in frames 40 and 220.

Figure 6 summarises tracking of a young girl running and jumping in front of a hand held, moving camera. Condensation starts to fail before the camera Shifts suddenly around frame 180. Mean Shift and Hybrid track well until frame 180, then fail due to high levels of both camera motion and target acceleration. SOK uses its structured search strategy to lock on to the girl at frame 180 and tracks her for the remainder of the sequence.

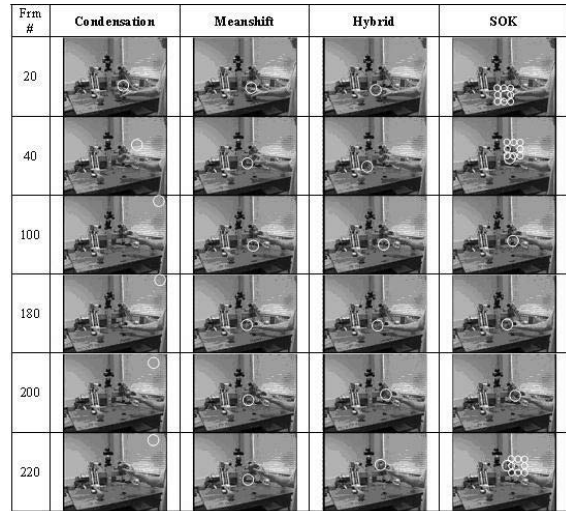


Figure 5. Tracking a hand-held ball through clutter.

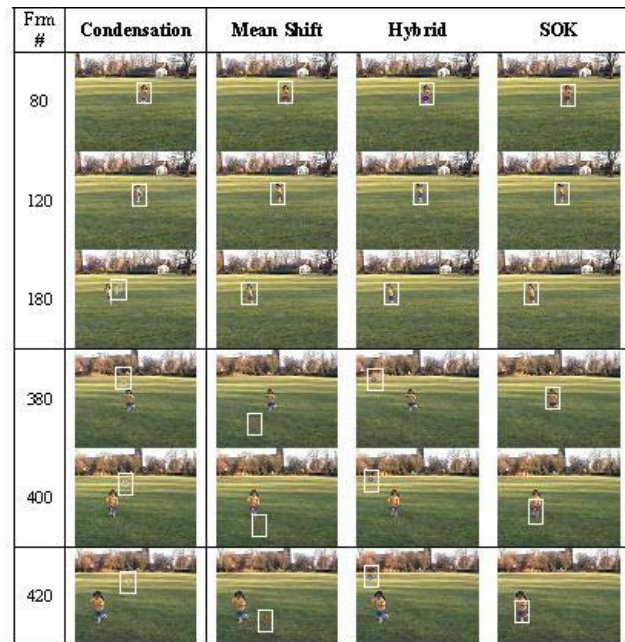


Figure 6. Tracking a girl at play.

5 Discussion

The Kernel Mean Shift tracker [9] is a robust and effective tracker with a very low computational cost. It performs well as long as the target object does not jump suddenly beyond its radius, or become occluded by an object with a similar model.

Condensation [1] outperforms Mean Shift during sudden object or camera motion, provided that a large enough particle set is employed. Increasing the number of particles used, however, quickly increases computational cost.

Maggio and Cavallaro's [8] Condensation/Mean Shift hybrid typically shows the performance expected of Condensation, but requires noticeably fewer particles, greatly reducing computational cost. The hybrid

tracker typically requires 80-90% fewer particles than regular condensation to achieve similar results [8].

The experimental evaluation presented here shows the proposed SOK filter to be more robust than the Condensation, Kernel Mean Shift, and the Hybrid trackers over a statistically significant set of (30) image sequences. SOK also provided more accurate tracking than Kernel Mean Shift and Condensation, with the Hybrid tracker performing comparably but requiring >5 times more particles. The SOK filter can easily be used at frame rates above 30fps, i.e. in real time, while the other three are quite costly computationally.

6 Conclusion

The ability of a particle set to represent a wide variety of distributions is both the main strength and primary weakness of particle filtering trackers. The particle set must sample widely enough that it can represent all reasonable alternatives in areas of ambiguity, but must not become diffuse, spreading across the image plane rather than clustering around the object of interest. When this happens particles tend to migrate towards local maxima in their evaluation function, becoming caught on clutter and losing track of the true target. A key issue in the design of particle filter-based trackers is how to manage the spread of the particle set to balance these conflicting requirements.

Maggio and Cavallaro's [8] hybrid tracker can be viewed as attempting to manage particle spread by alternately diffusing the particle set using Condensation and clustering it with kernel Mean Shift. Particle selection and initial posterior location is, however, managed by standard Condensation. If Condensation tends towards an incorrect local maximum, mean-shift will accelerate the process.

We have proposed a hybrid tracker that makes explicit the iterative diffuse-cluster structure implicit in Maggio and Cavallaro's work, only diffusing when necessary and then carpeting a fixed area around the prior with particles. The algorithm has been compared with Condensation, Kernel Mean Shift and Hybrid trackers. It provides the most robust results, with accuracy comparable to the Maggio and Cavallaro tracker [8], at the lowest computational cost.

7 Acknowledgements

This research was supported by a PhD Studentship awarded to Mr. A. Naeem by the Higher Education Commission of Pakistan.

8 References

[1] M. Isard and A. Blake, "CONDENSATION – conditional density propagation for visual tracking", *International Journal of Computer Vision*, 29(1) pp5-28, 1998.

[2] M. Pitt and N. Shephard, "Auxiliary particle filters" *J. Amer. Statist. Association*, 94(446), pp. 590-599, 1999.

[3] C. Musso, N. Oudjane, and F. LeGland, "Improving regularized particle filters", in *Sequential Monte Carlo Methods in Practice*, A. Doucet, J.F de Freitas, and N.J. Gordon, Eds. Springer-Verlag. New York, 2001.

[4] M. Isard and A. Blake, "ICondensation: Unifying low-level and high-level tracking in a stochastic framework", *Proc 5th European Conf. Computer Vision*, 1, pp 893-908, 1998.

[5] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line non-linear non-Gaussian Bayesian tracking," *IEEE Trans. Signal Processing*, 55 (2), pp. 174-188, Feb 2002.

[6] J. Deutscher, A. Blake, and I. Reid, "Articulated body motion capture by annealed particle filtering", *Proc. IEEE Conf. Computer Vision Pattern Recognition*, 2000.

[7] C. Chang and R. Ansari. "Kernel particle filter for visual tracking", *IEEE Signal Processing Letters.*, 12(3), pp. 242–245, 2005.

[8] E. Maggio and A. Cavallaro, "Hybrid particle filter and Mean Shift tracker with adaptive transition model", *Proc. Int. Conf. Acoustics, Speech, and Signal Processing* 2005.

[9] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking", *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(5), pp. 564–577, 2003.

[10] C. Yang, R. Duraiswami and L.S. Davis, "Efficient Mean Shift tracking through a new similarity measure", *Proc. IEEE Conf. Computer Vision Pattern Recognition*, 1, pp. 176-183, 2005.

[11] A.P. Leung and S. Gong, "Mean shift tracking with random sampling", *Proc. British Machine Vision Conference*, 2, pp. 729-738, 2006.

[12] <http://www.iis.ee.ic.ac.uk/~mpsha/ludwig/Video.html>. Visited on Sept. 15th 2006.

[13] <http://www.cs.nott.ac.uk/~tpp/Tracking/sequences.html>. Visited on Sept. 15th 2006.

[14] A Clark and C. Clark, "Performance Characterisation in Computer Vision – A Tutorial", <http://peipa.essex.ac.uk/benchmark/tutorials/essex/tutorial.pdf>. Visited on Sept. 15th 2006.

Image Segmentation Using an Active Contour Model

Byeong Rae Lee¹, YongKyu Kim², and Hyunchul Kang³

¹ Dept. Computer Science, Korea National Open University.

² Dept. Information and Communication Eng., Sungkyul Univ.

³ Dept. Information and Telecomm. Eng., University of Incheon

Email: brlee@knou.ac.kr

Abstract

In this paper, a novel image segmentation method based on active contour model is proposed. The primary requirement of this model is robustness to noise and brightness variation throughout the entire image. This model is based on Mumford-Shah functional for image segmentation and level sets. The boundary of an object in a given image is expressed as evolving curves that are zero level set of level set function ϕ . To make the zero level set of ϕ approach easily to the object boundary, the values of the function ϕ near the object boundary are controlled based on Laplacian of the image, direction of edge components and gradient of ϕ . The proposed method is applied to vehicle license plate images, and shows improved result compared to the existing methods.

Keywords: active contours, image segmentation, level sets

1 Introduction

Active contour models are widely used in many computer vision applications, including object segmentation. The basic idea in active contour models for object segmentation is to evolve a curve based on some constraints for a given image such that the contour converges toward the object's boundary[3-5].

Chan, et al. [1] proposed an active contour model based on techniques of curve evolution, Mumford-Shah functional for image segmentation, and level sets. This model can detect objects whose boundaries are not necessarily defined by gradient. When the brightness distribution of the background area over the image is not uniform, however, the contours cannot converge toward satisfactory boundary positions with Chan's method.

Non-uniform illumination condition like this is quite common in lots of applications, such as visual inspection in assembly lines, outdoor surveillance systems, etc. Lee, et al. [2] proposed an improved active contour model for these cases. The proposed method is based on modified Chan's energy functional, and shows significant improvement in non-uniform background brightness condition. The problem is that the energy functional in Lee's method includes a 2nd derivative term and it is too sensitive to small noise in background.

The purpose of this study is to provide a robust active contour model that can be used to segment objects in a given image. For curve evolution, level set method is applied to stop the evolution on the desired boundary.

2 Active Contour Models for Image Segmentation

In active contour models for object segmentation, an initial curve moves toward its interior normal until it reaches the boundary of an object. Let u_0 be a given image and C be a closed curve(active contour) in the image domain Ω . In the classical active contour model by Kass [3], a curve $C(s):[0,1] \rightarrow \mathbb{R}^2$ is moved to minimize the energy functional $J(C)$, where

$$J(C) = \alpha \int_0^1 |C'(s)|^2 ds + \beta \int_0^1 |C''(s)|^2 ds - \lambda \int_0^1 |\nabla u_0(C(s))| ds \quad (1)$$

The first two terms control the smoothness of the contour, and the third term attracts the contour toward the object in the image. It means that classical active contour models rely on the image gradient $|\nabla u_0|$, that acts as an edge-detector, to stop the curve evolution and they can detect objects with edges defined by gradient.

A general edge-detector can be defined by a positive and decreasing function g . For instance,

$$g(|\nabla u_0(x, y)|) = \frac{1}{1 + |\nabla G_\sigma(x, y) * u_0(x, y)|^p} \quad (2)$$

where $p \geq 1$ and $G_\sigma * u_0$ is a smoother version of u_0 . In practice, however, g is never zero on the edge, and the contour may not converge toward the object's boundary. For these cases, Chan and Vese[1] proposed an active contour model that does not

depend on edge calculation. Chan's model is based on Mumford-Shah functional for image segmentation and level sets. When we apply level set method to curve evolution, the unknown curve $C(t)$ that varies according to time t is replaced by the level set function $\phi(x, y, t)$. At any time t , $\phi > 0$ if any point (x, y) is inside C ; $\phi < 0$ if (x, y) is outside C ; and $\phi = 0$ if (x, y) is on C .

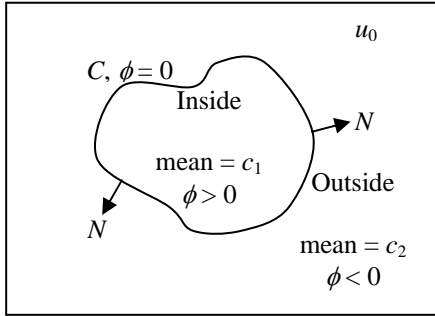


Figure 1: Curve C propagating in normal direction[1].

Suppose that an object is in a given image u_0 , as in figure 1. The values c_1 and c_2 are the averages of u_0 inside and outside of C respectively. Chan introduced an energy functional $F(c_1, c_2, C)$ defined by

$$F(c_1, c_2, C) = \mu \cdot \text{Length}(C) + \nu \cdot \text{Area}(\text{inside}(C)) + \lambda_1 \int_{\text{inside}(C)} |u_0(x, y) - c_1|^2 dx dy + \lambda_2 \int_{\text{outside}(C)} |u_0(x, y) - c_2|^2 dx dy, \quad (3)$$

where $\mu \geq 0$, $\nu \geq 0$, $\lambda_1, \lambda_2 \geq 0$ are fixed parameters. Chan rewrote equation (3) as a function of ϕ :

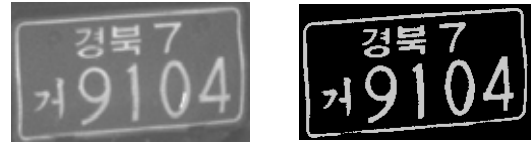
$$F_\varepsilon(\phi) = \mu \int_{\Omega} \delta_\varepsilon(\phi(x, y)) |\nabla \phi(x, y)| dx dy + \nu \int_{\Omega} H_\varepsilon(\phi(x, y)) dx dy + \lambda \int_{\Omega} |u_0(x, y) - c_1|^2 H_\varepsilon(\phi(x, y)) dx dy + \lambda \int_{\Omega} |u_0(x, y) - c_2|^2 (1 - H_\varepsilon(\phi(x, y))) dx dy \quad (4)$$

where H_ε is a regularized Heaviside function and δ_ε is the derivative of H_ε . Keeping c_1 and c_2 fixed, and minimizing F_ε with respect to ϕ , the associated Euler-Lagrange equation for ϕ is deduced as equation (5).

$$\phi_t = \delta_\varepsilon(\phi) \left[\mu \operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \nu - \lambda_1 (u_0 - c_1)^2 + \lambda_2 (u_0 - c_2)^2 \right] \quad (5)$$

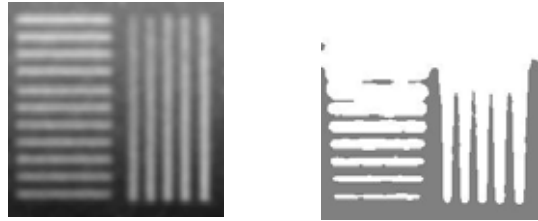
Chan's model shows good result when the brightness distributions are uniform both in background and foreground. In case of non-uniform brightness condition, however, the difference between c_1 and c_2 is not prominent enough for equation to lead contour

toward correct boundary position. Figure 2 is an example of the former case, while figure 3 and 4 is an example of the later case. In figure 2, overall brightness distribution is uniform, and all the characters are correctly segmented. In figure 3, on the other hand, brightness on the upper side of the image is brighter than that of lower side, and some of the stripes are mixed up together in the segmentation result. In figure 4, some of the characters are lost in the segmentation result because of the brightness difference between left and right part of the image.



(a) original image(u_0) (b) segmentation result

Figure 2: Segmentation result of Chan's model (test image-1)



(a) original image(u_0) (b) segmentation result

Figure 3: Segmentation result of Chan's model (test image-2)



(a) original image(u_0) (b) segmentation result

Figure 4: Segmentation result of Chan's model (test image-3)

To solve this problem, Lee, et al. [2] proposed a modified energy functional. Figure 5 shows a relationship among the image u_0 , the gradient ∇u_0 , and Laplacian Δu_0 , when a bright object is placed on a dark background. The value of Δu_0 is greater than 0 at the area near the boundary of dark background, and less than 0 at the area near the boundary of bright object. This idea can be realized as in equation (6).

$$D_\varepsilon(\phi) = \int_{\Omega} H_\varepsilon(\phi(x, y)) \Delta u_0 dx dy - \int_{\Omega} (1 - H_\varepsilon(\phi(x, y))) \Delta u_0 dx dy = \int_{\Omega} (2H_\varepsilon(\phi(x, y)) - 1) \Delta u_0 dx dy \quad (6)$$

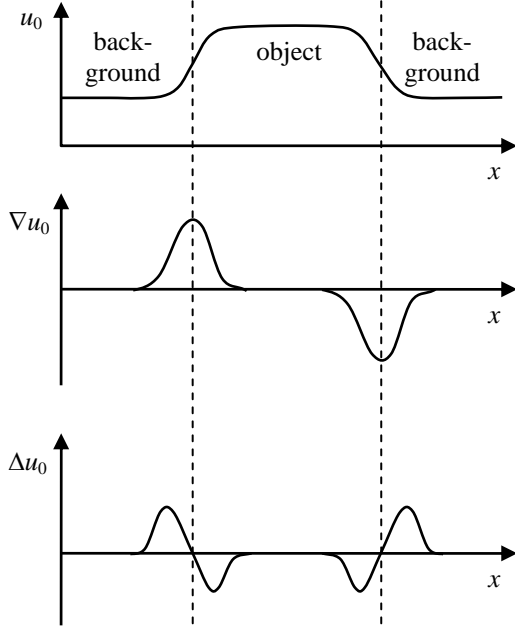


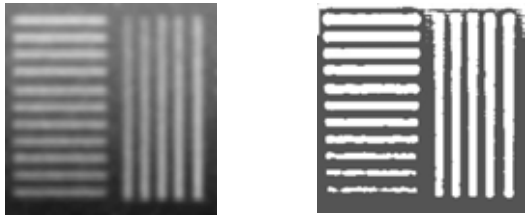
Figure 5: $|\nabla u_0|$ and Δu_0 of a given image u_0

The value of equation (6) will be it's maximum when the contours are positioned at the boundary of the objects. Based on this idea, Lee, et al, proposed a modified energy functional $F_L(\phi)$:

$$\begin{aligned}
 F_L(\phi) = & \mu \int_{\Omega} \delta_{\varepsilon}(\phi(x, y)) |\nabla \phi(x, y)| dx dy \\
 & + v \int_{\Omega} H_{\varepsilon}(\phi(x, y)) dx dy \\
 & + \lambda \int_{\Omega} |u_0(x, y) - c_1|^2 H_{\varepsilon}(\phi(x, y)) dx dy \quad (7) \\
 & + \lambda \int_{\Omega} |u_0(x, y) - c_2|^2 (1 - H_{\varepsilon}(\phi(x, y))) dx dy \\
 & - \eta_1 \int_{\Omega} (2H_{\varepsilon}(\phi(x, y)) - 1) \Delta u_0 dx dy
 \end{aligned}$$

The Euler-Lagrange equation to minimize $F_L(\phi)$ is given in equation (8).

$$\begin{aligned}
 \phi_t = \delta_{\varepsilon}(\phi) \left[\mu \operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - v - \lambda_1 (u_0 - c_1)^2 \right. \\
 \left. + \lambda_2 (u_0 - c_2)^2 + 2\eta_1 \Delta u_0 \right] \quad (8)
 \end{aligned}$$



(a) original image(u_0) (b) segmentation result

Figure 6: Segmentation result of Lee's model (test image-2)



(a) original image(u_0) (b) segmentation result

Figure 7: Segmentation result of Lee's model (test image-3)

When Lee's model is applied to the same image in figure 3, stripes are separated as shown in figure 6(b). Also, in figure 7(b), all the characters in test image-3 are segmented.

In spite of this improvement, Lee's model shows a weakness for images that contain noise, as in figure 8. This problem is caused by the Laplacian term in equation (8) that can dominate other terms when noise is added to the given image.



(a) original image(u_0) (b) segmentation result

Figure 8: Segmentation result of Lee's model (test image-4)

3 Proposed Active Contour Model

Chan's model and Lee's model show weakness in robustness against brightness change and noise respectively. In the area near the contour, we can include terms into ϕ to change ϕ appropriately. These terms are decided under the consideration on the way how to change ϕ according to the direction of ∇u_0 and $\nabla \phi$.

When the direction of ∇u_0 is similar to that of $\nabla \phi$ as in figure 9, the zero level set of ϕ should be moved to the edge. This can be implemented by increasing or decreasing ϕ according to the value of Δu_0 . If Δu_0 is positive at any (x, y) , this position is outside the object boundary, and ϕ should be decreased to push 0-level set of ϕ toward the boundary. Similarly, if Δu_0 is negative at any (x, y) , this position is inside the object boundary, and ϕ should be increased to push 0-level set of ϕ toward the boundary.

On the contrary, when the direction of ∇u_0 is similar to the opposite direction of $\nabla \phi$ as in figure 10, the zero level set of ϕ should be moved away from the edge. If Δu_0 is positive at any (x, y) , this position is outside the object boundary, and ϕ should be decreased to push 0-level set of ϕ away from the boundary. Similarly, if Δu_0 is negative at any (x, y) , this position is inside the object boundary, and ϕ should be increased to push 0-level set of ϕ away from the boundary.

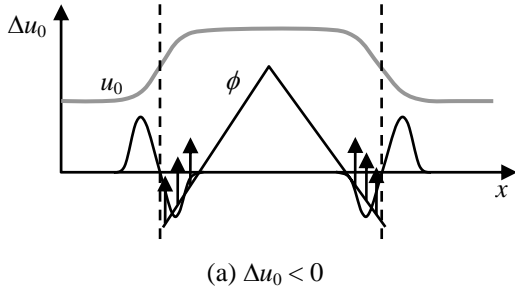
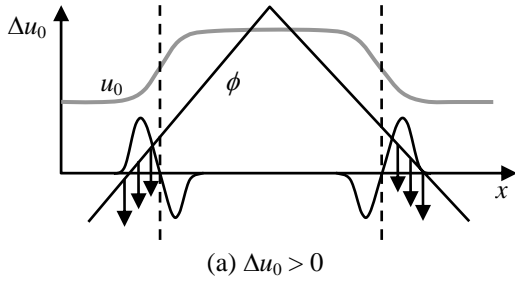


Figure 9. Modification of ϕ at any position where the direction of ∇u_0 is similar to that of $\nabla \phi$.

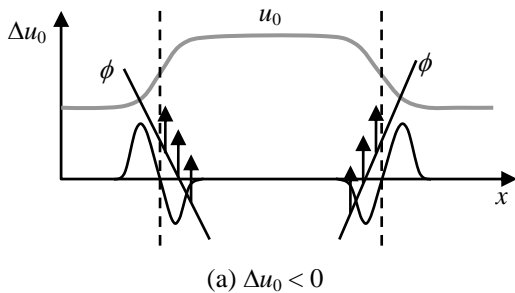
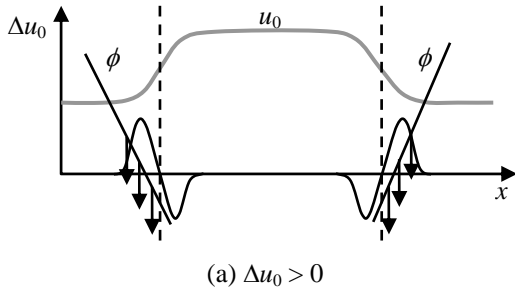


Figure 10: When the direction of ∇u_0 is similar to the opposite direction of $\nabla \phi$.

This concept can be implemented by modifying ϕ near its 0-level contour as in equation 9.

$$\begin{aligned} \phi_t = & \delta_\varepsilon(\phi) \left[\mu \operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \nu - \lambda_1 (u_0 - c_1)^2 \right. \\ & + \lambda_2 (u_0 - c_2) + 2\eta_2 \Delta u_0 \\ & \left. - \gamma \frac{\Delta u_0}{|\Delta u_0|} \nabla u_0 \cdot \frac{\nabla \phi}{|\nabla \phi|} \right] \end{aligned} \quad (9)$$

The algorithm is implemented in two phase. In the 1st phase, equation (8) is applied to a given image to find initial contours. In the 2nd phase, equation (9) is applied to the result of phase-1 to refine the result.

4 Experimental Result

To evaluate the performance of proposed algorithm, Chan's method[1], Lee's method[2], and the proposed algorithm are compared by applying those algorithms to several vehicle licence plates images. The values of parameters are as follows:

$$\mu = 0.01 \max(u_0)^2, \nu = 1, \lambda_1 = \lambda_2 = 1,$$

$$\eta_1 = 1000, \eta_2 = 100, \gamma = 0.1$$

Figure 11 shows the segmentation results for an image that has relatively uniform background brightness and contains only small amount of noise. All of the three method segments characters correctly.

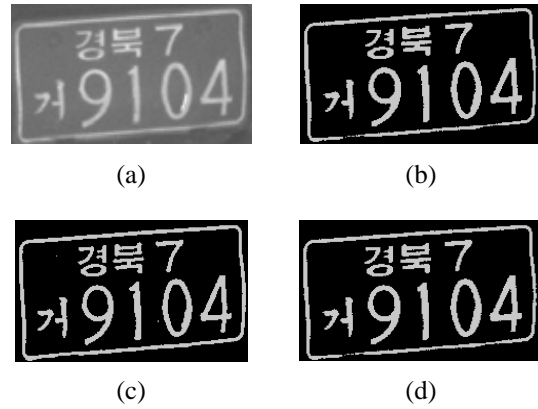


Figure 11: Image Segmentation Results(test image-1), (a) Original Image, (b) Chan's method, (c) Lee's method, (d) proposed method

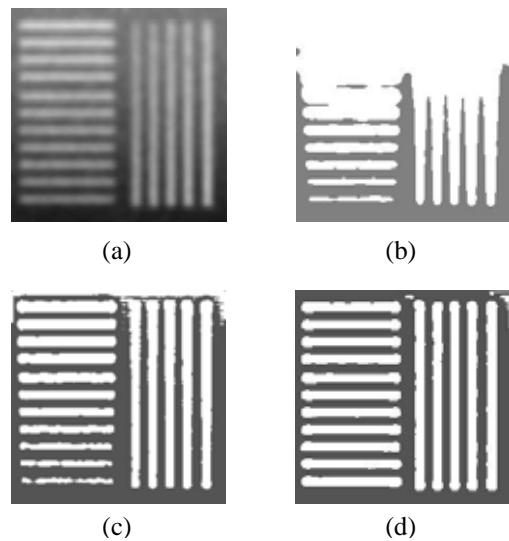


Figure 12: Image Segmentation Results(test image-2), (a) Original Image, (b) Chan's method, (c) Lee's method, (d) proposed method

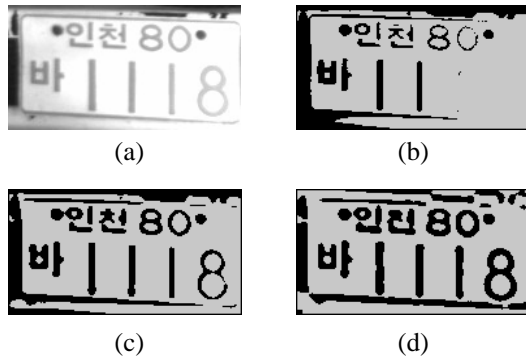


Figure 13: Image Segmentation Results(test image-4), (a) Original Image, (b) Chan's method, (c) Lee's method, (d) proposed method

The original image in figure 12 and figure 13, however, has non-uniform background brightness, and Chan's method fails to segment objects in the given images. Both Lee's and proposed method segment objects in the images, and the proposed method shows slightly better results.

The original image in figure 14, however, has non-uniform background brightness and contains noise. Chan's method (figure 14(b)) fails for this image because of the non-uniform background brightness. Lee's method (figure 14(c)) shows weakness against noise. Proposed method shows better result compared to previous methods.

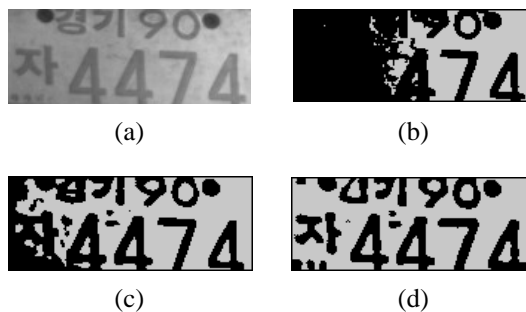


Figure 14: Image Segmentation Results(test image-3), (a) Original Image, (b) Chan's method, (c) Lee's method, (d) proposed method

5 Conclusion

Image segmentation is a fundamental and vital in computer vision systems. In this paper, an image segmentation algorithm that can be applied to pattern recognition, visual inspection, visual surveillance, etc is proposed.

In many practical applications, brightness distribution over the image is not uniform. More over, the image may contain noise. The active contour model proposed in this paper is based on level-set theory framework to achieve image segmentation for these cases. Function ϕ is controlled to move 0-level set to object's boundary.

Compared to Chan's [1] and Lee's [2] algorithm, proposed method shows improved result. In particular, proposed method shows robustness in non-uniform illumination and noisy environment.

6 References

- [1] Tony F. Chan, Luminita A. Vese, "Active contours without edges," *IEEE Trans. Image Processing Proceeding*, Vol.10, No.2, pp. 266–277, 2001.
- [2] B.R. Lee, A.B. Hamza, and H. Krim, "An active contour model for image segmentation: A variational perspective," *Proc. ICASSP2002*, pp. II-1585–II-1588, 2002.
- [3] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.*, vol. 1, pp. 321–331, 1987.
- [4] V. Caselles, R. Kimmel, G. Sapiro, "Geodesic active contours," *Int. J. Comput. Vis.*, vol.22, no.1 pp.61–79, 1997.
- [5] N. Paragios and R. Deriche, "Geodesic active contours and level sets for the detection and tracking of moving objects," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp.1–15, 2000.
- [6] J. A. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision and Materials Science*, 2nd Ed. New York: Cambridge University Press, 1999.

Joint Outliers and Principal Component Analysis

Georgy Gimelfarb, Alexander Shorin, and Patrice Delmas

Dept. of Computer Science, University of Auckland, P.B. 92019, Auckland, New Zealand.

Email: g.gimelfarb@auckland.ac.nz, al@cs.auckland.ac.nz, p.delmas@auckland.ac.nz

Abstract

Today PCA is very popular in various areas of pattern recognition, computer vision, and image retrieval, so its robustness to interfering data variations, e.g. nuisance image detail, is of a great practical value. This paper proposes to build a robust PCA by “soft masking” artefacts in an image database. Data variations with respect to a linear subspace of PCs are described by the mixture of Gaussian noise and uniformly distributed outliers. Soft masks of outliers for data items are produced by an iterative Expectation-Maximisation (EM) algorithm. Then, the characteristic PCs are formed for the masked database. Our experiments show the proposed JOPCA overcomes drawbacks of some conventional robust PCA techniques.

Keywords: pattern recognition, face recognition, robust PCA, Expectation-Minimization

1 Introduction

Statistical dimensionality-reduction techniques have proven to be useful in many pattern recognition, computer vision, and image retrieval applications. They are particularly important for image recognition and learning tasks where “the curse of dimensionality” [1] makes it impossible to deal with the problem space of an untransformed pixel domain. Principal component analysis (PCA) is one of the most significant and universal tools among them both historically [2] and practically [3].

PCA works well on datasets where images were regularized with respect to many spatial domain parameters including positioning, rotation, lighting, size, background, and so on [4]. However, some interfering features cannot be eliminated completely due to practical considerations. For example, frequent visual occlusions in face recognition are caused by facial hair, glasses, scarves, and so on. A pixel affected by one of these unwanted artefacts can be defined as an *outlier*. A more formal definition is presented in Section 3.

Outliers in images are detrimental to recognition accuracy due to two peculiarities of PCA. First, if a subject’s image is corrupted with outliers, the ability to classify subsequent “clean” images of the



(a) Original image and its reconstructions from 1, 3 and 5 PCs



(b) Corrupted image and its respective reconstructions

Figure 1: Effects of outliers on a lossy PCA recovery.

same subject will predictably suffer. As an example, let us consider the lossy reconstruction of an image from its 1, 3 and 5 most significant PCs as displayed in Fig. 1. If no images were occluded, the computed approximation appears as expected (Fig. 1(a)). However, if an image was corrupted with visible nuisance features (Fig. 1(b)) its reconstruction displays their evident presence even when only one PC was used.

Second, due to the nature of PCA, artefacts in even a single image will pervade all other images in the database. In Fig. 2 just as in the previous example one of the images is corrupted. The respective pixels in the other images will also be contaminated with the introduced error once they are reconstructed from their PC scores. One can notice the silhouettes of the glasses and moustache on the image which did not contain them originally. Simultaneously, the retrieval accuracy will decline.

Not only occlusions should be treated as outliers. Background is likely to considerably vary in different facial images and thus its inclusion into PCA is equally a bad strategy. At present the standard approach in building face databases is to crop images in order to completely eliminate their background. However, if background pixels are regarded as outliers they could be eliminated automatically, like occlusions. This paper introduces an algorithm for eliminating



Figure 2: Original “clean” image and its reconstructions from 1, 3 and 5 PCs when one other image in the database is corrupted with noise. The effect of noise is noticeable even when 5 PCs were used.

outliers. The suspicious pixels are simply masked out before PCA is conducted. The computed mask can eliminate both occlusions and arbitrary backgrounds. Our approach adopts *soft masking* where each pixel is weighed by a corresponding mask entry ranging from 0 to 1. The masks are found with an Expectation-Maximisation (EM) algorithm which is much faster and more flexible than more conventional available methods. In addition, our approach strictly enforces the orthogonality of the transformed space¹, and hence it is more suited for large databases where the orthogonality is essential to guarantee successful scalability.

2 Background

Our data model extends the conventional PCA model described in brief below. A reader familiar with PCA can skip Sections 2.1 and 2.2.

2.1 Generalised PCA

Let G be a $p \times n$ matrix of n images, each of size of p pixels, such that each element g_{ij} is the value of the i -th pixel in the j -th image in the mean-deviation form. Each image is a column g_j of the matrix G . The size $p = |g_j|$ is a product of horizontal and vertical resolutions of the image. All images have identical dimensionality.

Generalised PCA (GPCA) starts with finding a covariance matrix S of G : $S = \frac{1}{n-1}GG^T$ where T denotes transposition. Since S is symmetric, its eigenvectors form an orthogonal basis. Let E be a $p \times p$ matrix of eigenvectors-columns \mathbf{e}_i sorted in the descending order of the magnitude of their eigenvalues λ_i : $E = [\mathbf{e}_1 \ \mathbf{e}_2 \ \dots \ \mathbf{e}_p]$. These eigenvectors, called principal components (PC), are used to linearly transform the matrix G into the matrix Y of its PC scores: $Y = E^T G$. Although algebraically each element (PC score) of Y is a linear combination of pixels in G , it is also viewed

¹This property is sometimes dropped in the known algorithms to increase their computational efficiency.

geometrically as a projection of an image g_j represented as a point in a p -dimensional space onto a one-dimensional subspace defined by the corresponding eigenvector \mathbf{e}_k . Each eigenspace k in E explains the $\lambda_k/\text{trace}(S)$ part of the total variance in G . Since E is sorted in the descending order of λ , the score y_k stores more or equal information about G than y_{k+1} for $1 \leq k < p$. Consequently, in a lossy GPCA the first m PC scores, $1 \leq m < p$, corresponding to the eigenspaces with the larger λ are required to guarantee the minimum error in reconstructing G from any m PCs. The approximate matrix \hat{G} is computed from a subset E_m of p eigenvectors in E as $\hat{G} = (E_m^T)^{-1}Y$. In most of applications its pseudoinverse E_m^+ is used: $\hat{G} = (E_m^T)^+Y$. In the case of a lossless PCA $E_m = E$ and $\hat{G} = G$.

2.2 Tractable PCA

GPCA finds the complete eigenspace E which is grossly redundant and likely to be intractable. Although the calculation of eigenspaces can find the complete set of p eigenvectors, only a few of them, j , $1 \leq j < p$, with the sizeably non-zero eigenvalues λ_k , $k = 0, 1, \dots, j$, are significant for the analysis. The rest of the eigenspaces capture almost zero or zero variance and hence do not serve any useful purpose. For a database of images with an $m \times m$ resolution, GPCA requires the memory of order of m^4 to represent the corresponding covariance matrix. Not only this is a very large data structure, what is worse, finding eigenvectors for so large matrices is computationally infeasible.

Tractable methods alternative to GPCA that successfully compute eigenspaces by excluding the brute force steps of GPCA made it possible to use PCA for computer vision tasks [2]. Since $G^T G \mathbf{e} = \lambda \mathbf{e}$ and $G G^T G \mathbf{e} = \lambda G \mathbf{e}$, a subset $E' \subset E$ of all eigenspaces of $G G^T$ could be obtained as $E' = G \tilde{E}$ where \tilde{E} is the matrix of the eigenvectors of $\tilde{S} = G^T G$. It can be shown that the eigenvector $\mathbf{e}_j \in E'$ if and only if $\lambda_j \neq 0 \ \forall \mathbf{e}_j \in E$. As a result, only eigenspaces that capture non-zero variance in the original data are found by this method. Also, the memory requirements for \tilde{S} are only of the order of n^2 , making the eigenanalysis tractable for reasonably large image databases. The rest of the analysis, i.e. computing projections Y and approximations \hat{G} , is performed just as in GPCA.

2.3 Robust PCA

PCA does not deal successfully with noisy data. As a result, more robust to noise algorithms have been proposed [5, 6]. Unfortunately, they cannot be extended onto high dimensional data sets typical

to image processing. Only very recently a handful of methods which could be applied with some success to computer vision problems have appeared. One of the proposed robust PCA techniques [7] solves the following optimisation problem: $\text{error} = \min_{(E;V)} \varepsilon(E;V)$ where

$$\varepsilon(E;V) = \sum_{j=1}^n \left[V_j \left(\sum_{i=1}^p \left(g_{ip} - \sum_{k=1}^m e_{ik} y_{jk} \right)^2 \right) + \eta(1 - V_j) \right]$$

Here, m is the number of PCs used and the binary variable $V_j \in \{0, 1\}$ indicates whether an image j is discarded as an outlier or not. The term $\eta(1 - V_j)$ is a penalty to avoid the global optimum where all images are rejected. Unfortunately, the main problem with the approach in [7] is that entire images would be rejected which does not make it particularly practical.

A more interesting recent approach in [8] moves from treating entire images as outliers to dealing with the pixel domain. It optimizes the following error function:

$$\text{error} = \min_{E, Y, \mu, \sigma} \sum_{i=1}^p \sum_{j=1}^n \rho(g_{ij} - \mu_i - \sum_{k=1}^m e_{ik} y_{jk}, \sigma_i)$$

where m is as above, μ_i is the average of the i -th pixel across all images, $\rho(w, \sigma_i) = \frac{w^2}{w^2 + \sigma_i^2}$ is the Geman-McClure error function, and the parameter σ_i follows from the assumed normal distribution of noise in every reconstructed pixel: $\text{error}_{ip} \sim \mathcal{N}(0, \sigma_i^2)$.

Although the approach in [8] successfully created a robust system, it has serious drawbacks. First, the learned vectors of E are not generally orthogonal which will be a major obstacle to scalability of the method since more vectors will be required to capture the same amount of variance comparing to regular PCA². Secondly, the error is minimised by using gradient descent (GD) in the error space that consists of the four sets of parameters (E, Y, μ, σ) . The authors in [8] claim that the computational cost of each iteration is $\mathcal{O}(pnm)$, where m is the number of robust PCs³. Not surprisingly that for a 256-image database with 20 robust PCs the algorithm took hours to converge. Finally, what is most important, an addition of extra images to the database requires a complete recomputation of the learned space.

Other known robust PCA methods, for example, in [9, 10, 11], deal with either pixels or images as

²The lack of orthogonality, strictly speaking, means that their robust method is not a PCA.

³Note that vectors in E are not, strictly speaking, the PCs since they lack orthogonality.

outliers; however they all have very similar drawbacks. Our approach introduced below detects outliers at the pixel level, but in contrast to the previous algorithms, avoids successfully all three aforementioned problems.

3 Joint outliers and PC analysis

3.1 Error model

The reconstruction of an original image is inevitably imprecise if a lossy transformation (such as a lossy PCA) is used: $g_{ij} = \hat{g}_{ij} + \varepsilon_{ij}$ where $\varepsilon_{ij} \in [(-2^b + 1), (2^b - 1)]$ denotes the pixel-wise error, b is the number of greyscale bits used to encode the image, and \hat{g}_{ij} is a reconstructed signal defined as $\hat{g}_{ij} = \sum_{j=1}^c e_{ij}^+ y_{jk}$. Here, $1 \leq c \leq n$, $e_{ij}^+ \in E^+$, and $y_{jk} \in Y$ is the score of the j -th PC for the k -th image.

We assume that the error ε_{ij} may be caused by either *noise* or an *outlier* and in both cases errors are statistically independent in the pixels and images. This dichotomy forms the error model for our approach. Informally, the objective is to include pixels generating noise in PCA⁴, while to detect and exclude outlying pixels. Formally, the probability distribution of ε_{ij} over all images is specified as the mixture of a normal zero-centred distribution for noise :

$$\mathcal{N}(\varepsilon|0, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-\varepsilon^2}{2\sigma^2}\right)$$

with a fixed but unknown variance σ^2 , and a uniform distribution for outliers:

$$\Pr(\varepsilon) = \alpha\mathcal{N}(\varepsilon|0, \sigma) + (1 - \alpha)u_b$$

where α is an unknown prior probability of non-outliers and $u_b = 1/(2^{b+1} - 1)$ is the probability of uniformly distributed outliers.

Experimental results suggest that this simple hypothesis is likely to be accurate due to a clearly observable Gaussian shape of empirical distributions for small errors but levelled long tails for the large ones. This provides us with a decision rule for delineating noise from outliers: to mask out pixels which errors are significantly larger than one might expect from a normal noise. The like rule based on a maximum likelihood estimator and an EM-based evaluation strategy is introduced below.

3.2 Soft masking of outliers

The underlying idea is to detect outliers, mask them out, and restrict PCA only to relevant pixels

⁴Since the noise is an inherent feature of lossy transformations

in every image. Let a mask M exist for the data matrix G such that every entry m_{ij} of M applies to the corresponding pixel g_{ij} . One approach, the binary mask, $m_{ij} \in \{0, 1\}$, simply cuts out pixels with zero mask value and retain ones with the unit mask value, i.e. converts every image g_j into the masked image g_j° . The computation of all pixel-wise and pairwise statistics to obtain the covariance matrix \hat{S} involves the masks in an obvious way: M works as a "mask-out", namely, a multiplication by M *erases* unwanted pixels. Hence, tractable PCA is easily adapted to the use of such masked images.

Actually, the mask M has to be evaluated from the available data using the assumed noise model. To fit the EM framework [12], let us replace the binary mask with a soft one such that the smaller the value $m_{ij} \in [0, 1]$, the higher the plausibility that g_{ij} is an outlier. A single value m_{ij} represents a degree of responsibility of a pixel for both the categories "noise" and "outlier" simultaneously. Initially all entries in M are initialised to 1, so no information is filtered out: all pixels are 100%-responsible for inclusion to PCA and 0%-responsible for elimination. As the PCs and the mask are recalculated iteratively, the responsibilities m_{ij} change.

The first step in calculating G° , the masked version of G , is to compute \bar{G} containing vectors of means \bar{g}_i for each pixel i :

$$\bar{g}_i = \frac{\sum_{j=1}^n g_{ij} m_{ij}}{\sum_{j=1}^n m_{ij}}; \quad i \in \{1, \dots, p\}$$

G° is obtained in the mean-deviation form with $g_{ij}^\circ = (g_{ij} - \bar{g}_j) m_{ij}$.

Next, the covariance matrix S° is calculated with the entries

$$s_{ij}^\circ = \frac{\sum_{k=1}^n g_{ik}^\circ g_{jk}^\circ m_{ik} m_{jk}}{\sum_{k=1}^n m_{ik} m_{jk}}; \quad i, j \in \{1, \dots, p\}$$

where the sum is taken over the i -th and j -th pixels of all images. Then the matrix S° is processed as usually by PCA.

To re-evaluate M , the soft mask values are formed at every iteration t as expected responsibilities of g_{ij} for the current reconstruction noise $\varepsilon_{ij}^{[t]}$:

$$\begin{aligned} m_{ij}^{[t+1]} &= \frac{\alpha^{[t]} \mathcal{N}(\varepsilon_{ij}^{[t]} | 0, \sigma^{[t]})}{\alpha^{[t]} \mathcal{N}(\varepsilon_{ij}^{[t]} | 0, \sigma^{[t]}) + (1 - \alpha^{[t]}) u_b} \\ \alpha^{[t+1]} &= \frac{1}{np} \sum_{i=1}^p \sum_{j=1}^n m_{ij}^{[t]} \\ \sigma^{[t+1]} &= \sqrt{\frac{\sum_{i=1}^p \sum_{j=1}^n (\varepsilon_{ij}^{[t]})^2 m_{ij}^{[t]}}{\sum_{i=1}^p \sum_{j=1}^n m_{ij}^{[t]}}} \end{aligned}$$

The mask is calculated iteratively until the local minimum of the $L2$ error function over G is found. The only parameters which need to be estimated experimentally are σ and α determining the soft decision for g_{ij} of being a noise or an outlier.

3.3 Computational considerations

As was mentioned in Section 2, the current state-of-the-art systems have at least three major drawbacks: non-orthogonality, the use of GD-based optimisation, and the need to recompute the entire space after updating the dataset. Compared to them, our system is better scalable, faster, and more flexible.

First, the orthogonality of PCs is preserved in our method. Therefore, the efficient data compression of PCA is guaranteed, and this makes the system scalable easier than its predecessors.

Second, we need not use GD to solve the optimisation problem and hence avoid a number of computational issues connected to a GD paradigm: slow convergence and locally optimal solutions which are too far from the desired optimum. Most of current robust PCA methods for computer vision minimise the error of a lossy PCA-based reconstruction of images in a multidimensional parameter space, as e.g. in [8] (see Section 2.3). In our case the parameter space is much simpler with only two unknown parameters α, σ to be estimated by the iterative EM-algorithm, where the mask the mask M is dependent analytically on these parameters. The previous approaches had to use GD for searching for an acceptable local minimum of error in spite of all its inherent computational limitations such as slow convergence and high chances of getting stuck with some dramatically suboptimal solution. Our approach deals with a more global search of the error space because iterative readjustments of the masking weights in M result in pretty large movements across the parametric space, roughly in the direction of the global minimum as justified by the underlying EM algorithm [12]. Consequently the computations are much faster. The improvement comes from a dramatic reduction of the number of iterations required to find an acceptable suboptimal solution.

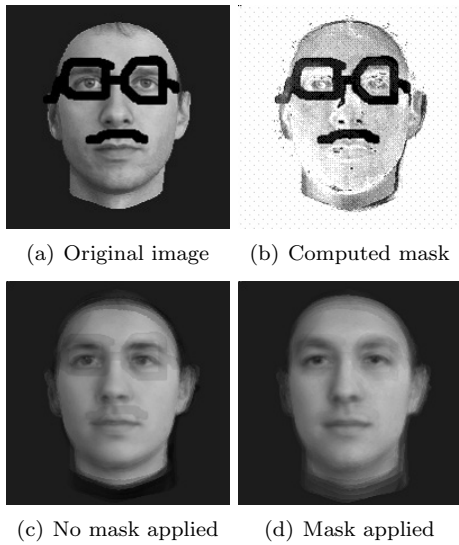


Figure 3: The effect of the mask on the reconstruction of an image with outliers

Finally, the method does not require a complete recomputation of the mask if additional images are added to the database. We only need to recompute the PC space of the updated database, with recomputation of the mask left as a periodic off-line task. This is a significant advantage over all other robust methods published up to date which would require a complete rerun.

4 Experimental results

A subset of the MIT dataset [13] was selected consisting of two images with different lighting for each of the 10 people (20 grayscale images in total; 8 bits per pixel). All images are normalised and have the resolution of 200 by 200 pixels. Our algorithm was implemented in accord with its description so that each pixel in G had its own unique mask. Experiments suggest that the best results can be achieved when the number of PCs $k \ll n$. In fact it appears that filtering is most effective when $k \rightarrow 1$. Specifically, fig. 3 displays the success of JOPCA with $k = 1$. Clearly, the occlusions created by the glasses and moustache appear to have been successfully removed. More experiments are under way at present for other values of k .

An interesting consequence of calculating the unique soft mask m_{ij} for every pixel is that some of the legitimate signal variations are filtered out as well. The mask in Fig. 3(b) shows that some relevant detail is removed from the face. Although this has not caused a problem for this particular image, other images in the same dataset had some additional *auxilliary* noise in their reconstructed images. Fig. 4 demonstrates this undesired effect.

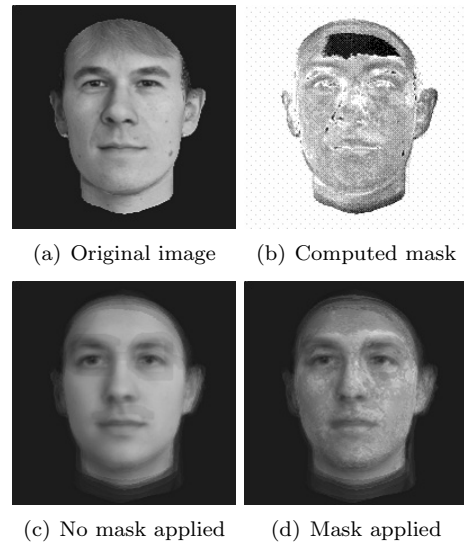


Figure 4: The effect of the mask on the reconstruction of an image with no outliers

We expect that less auxilliary noise will be generated by JOPCA if we move away from calculating a unique m_{ij} for every pixel. Some possible refinements are discussed below, in Section 5.

5 Future work and conclusions

Our initial experiments confirm that the proposed JOPCA works quite effectively and is computationally much more attractive than other known robust methods. Unfortunately, we cannot compare the performance of algorithms experimentally since other authors have not published information which will make such benchmarking possible. It is our objective though to make the detailed results obtained from running our method available publicly.

Our current work-in-progress is to refine our approach further and at present we are in the process of comparing the following methods for computing the soft mask:

Uniform image mask: One of the ways of preventing the appearance of auxilliary noise in reconstructed “clean” images is to calculate one uniform mask which could be applied to all images uniformly. Thus, entire “difficult” areas where occlusions such as glasses appear frequently would be ignored by PCA.

Region-based image mask: Another possibility is to implement a region-based filter. Hence more attention will be paid to error pervasively present in the area rather than error randomly generated by a lossy PCA. The size of the frame needs to be selected by a separate procedure.

Uniform region-based image mask: The idea behind this is to see whether the expected advantages of either could be combined into one robust implementation.

The results reported so far were obtained from running the algorithm on the MIT dataset [13] where all images have their background cropped out. We are also in the process of creating a working model which should demonstrate how well the algorithm works with non-trivial background. So other available databases, such as the Yale dataset [14], will be considered in the future work.

Another pending project is to evaluate the algorithm with more realistic outliers (not just thick artefacts used in the initial experiments).

Finally, we are looking at evaluating JOPCA for image retrieval purposes. At this stage, the challenges are to assess to what extent the elimination of outliers provided by JOPCA and the additional noise introduced by JOPCA affect the accuracy of image retrieval.

In total, we have presented a novel robust PCA model, JOPCA, and have demonstrated that it is much faster and more flexible than other available alternatives. It successfully deals with noisy input data which makes it important as a practical platform for many computer vision applications. Our new model successfully overcomes the three significant drawbacks of many of contemporary algorithms: it offers optimal compression due to preserving the orthogonality of the PC space, rapid conversion, and flexibility to online updating.

References

- [1] R. Bellman, *Adaptive Control Processes*. Princeton University Press, 1961.
- [2] M. Turk and A. Pentland, "Face recognition using eigenfaces," in *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR'91), June 1991, Lahaina, Maui, Hawaii*, pp. 586–591, IEEE CS Press, 1991.
- [3] K. Delac, M. Grgic, and S. Grgic, "Independent comparative study of pca, ica, and lda on the feret data set," *Int. J. Imaging Systems and Technology*, vol. 15, no. 5, pp. 252–260, 2005.
- [4] W. Zhao, R. Chellappa, P. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Computing Surveys*, vol. 35, no. 4, pp. 399–458, 2003.
- [5] N. Campbell, "Robust procedures in multivariate analysis 1: Robust covariance estimation," *Applied Statistics*, vol. 29, no. 3, pp. 231–237, 1980.
- [6] F. Ruymgaart, "Robust principal component analysis," *J. Multivariate Analysis*, vol. 11, no. 4, pp. 485–497, 1981.
- [7] L. Xu and L. Yuille, "Robust principal component analysis by self-organizing rules based on statistical physics approach," *IEEE Transactions on Neural Networks*, vol. 6, no. 1, pp. 131–143, 1995.
- [8] F. De la Torre and M. Black, "Robust principal component analysis for computer vision," in *Proc. Eighth Int. Conf. on Computer Vision (ICCV'01), July 7–14, 2001, Vancouver, British Columbia, Canada*, vol. 1, pp. 362–369, IEEE CS Press, 2001.
- [9] J. Karhunen and J. Joutsensalo, "Generalizations of principal component analysis, optimization problems, and neural networks," *Neural Networks*, vol. 8, no. 4, pp. 549–562, 1995.
- [10] M. Black and A. Rangarajan, "On the unification of line processes, outlier rejection, and robust statistics with applications in early vision," *Int. J. Computer Vision*, vol. 19, no. 1, pp. 57–91, 1996.
- [11] R. Rao, "An optimal estimation approach to visual perception and learning," *Vision Research*, vol. 39, no. 11, pp. 1963–1989, 1999.
- [12] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001.
- [13] MIT face database, accessed 24 Aug 2006, <http://vismod.media.mit.edu/pub/images/>
- [14] YALE face database, accessed 20 Sept 2006, <http://cvc.yale.edu/projects/yalefaces/yalefaces.html>

Integrated Test Pattern Generator and Measurement Algorithm for Colour Compression Artefacts in Ubiquitous Colour Spaces

G. A. D. Punchihewa, D. G. Bailey, and R. M. Hodgson

Institute of Information Sciences and Technology, Massey University, New Zealand

Email: g.a.punchihewa@massey.ac.nz

Abstract

This paper presents an environment to evaluate image codecs for the colour bleeding artefacts objectively. It is difficult to detect and measure individual artefacts in coded images. A synthetic random colour test pattern generator and a colour gamut transfer algorithm are developed to emphasise and measure colour bleeding artefacts due to image compression. The performances of a JPEG and a JPEG2000 codec implementations are compared in colour reproduction for television colour gamut. Both types of codecs show an increasing level of colour bleeding artefacts with increasing compression ratio. The objective artefact measures can be used in the image codec development process, in parameter optimisation of codec performance and in selecting a codec for a given application. Artefact metrics can also be used to select suitable parameters for video codecs while creating video streams for the Internet applications and in any multimedia application in general.

Keywords: colour bleeding, image compression, image artefacts, objective assessment, image quality, artefact metric, test pattern, colour space, colour gamut.

1 Introduction

In digital television broadcasting, video streaming and other multimedia communications, image and video are the dominant components. With limited communication bandwidth and storage capacity in terminal devices, it is necessary to reduce data rates using digital codecs. The techniques and quantisation used in image and video compression codecs introduce distortions known as artefacts. *The Digital Fact Book* defines artefacts as “particular visible effects, which are a direct result of some technical limitation” [1].

High levels of compression result in undesirable spurious features and patterns, and incorrect colours in the reconstructed image; these are the artefacts defined above. Image compression schemes may result in colour errors in addition to the blockiness, blur, contouring and ringing artefacts also found in coded images [2]. We have developed static test patterns and objective artefact metrics for blockiness, blur, ringing and colour bleeding artefacts in coded images [3, 4, 5], so blockiness, blur, and ringing effects will not be considered further in this paper.

JPEG2000 is an image compression standard based on the use of wavelets. It is gaining popularity because it delivers higher compression than JPEG for a given quality. It uses the complete image data at once in processing to obtain the frequency domain representation. JPEG is an image compression standard which has been common use over a longer time than JPEG2000. However, very little research has been done to benchmark and compare these two codecs for colour artefacts. JPEG has been in use for

compression of still images in video and television production facilities.

For many years, broadcasting engineers have been using standard colour bar test patterns for testing and adjustment of analogue colour television and video systems [6]. Analogue information is transformed into the hue-saturation-luminance colour space before it is transmitted to the viewers [7]. In analogue image and video systems, subjective assessments are made on preview monitors and objective assessments on measuring instruments such as a vectorscope. These enable an evaluation of perceptual quality as well as provide accurate and swift measurements. The synthetic colour bar test pattern used for analogue quality evaluation does not stress the codecs and does not provide suitable content for the measurement of colour artefacts in digital image and video systems [8]. Traditional vectorscope or waveform monitors do not provide assistance in making objective measurements on such codecs.

The approach in this paper is to use the full-referenced technique [9]; this involves the comparison of the reconstructed test pattern with the original test pattern. A random synthetic test pattern having known spatial distributions of coloured pixels will supplement the earlier static test pattern pool [3, 4, 5]. This paper demonstrates the concept of colour artefact assessment for a given colour space of luminance, hue and saturation and the colour gamut of PAL television colour system by performing a comparative study of coding colour artefacts due to the use of block processed discrete cosine transform (DCT) and wavelets in digital codecs. The procedure and the environment can be applied to any colour space or colour gamut and any colour image or video codec.

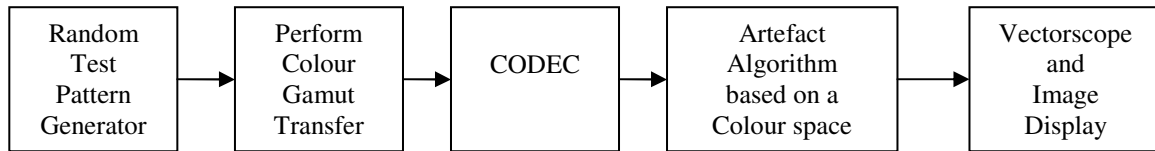


Figure 1: Block diagram of the integrated test pattern generator and colour artefact measurement environment

Table 1. Different colour spaces for common image and video codecs

Colour Space	Luminance	Colour Difference Signal	Colour Difference Signal
PAL	$Y = .299R + .587G + .114B$	$U = -.147R - .289G + .436B$	$V = .615R - .515G + .100B$
NTSC	$Y = .299R + .587G + .114B$	$I = .596R - .274G - .322B$	$Q = .211R - .523G + .311B$
SECAM	$Y = .299R + .587G + .114B$	$R - Y$	$B - Y$
JPEG2000	$Y = .299R + .587G + .114B$	$C_b = -.16875R - .33126G + .500B$	$C_r = .500R - .41869G - .08131B$
JPEG2000 lossless	$Y = .25R + .5G + .25B$	$U = R - G$	$V = -G + B$
CCIR 601	$Y = .257R + .504G + .098B + 16$	$C_b = -.148R - .291G + .439B + 128$	$C_r = .439R - .368G - .071B + 128$
CIE XYZ	$X = .4306R + .3415G + .1784B$	$Y = .2220R + .7067G + .0713B$	$Z = .0202R + .1295G + .9394B$

2 Methodology

In our previous work of colour artefact evaluation and test pattern generation, a static colour test pattern was designed to evaluate coding colour bleed artefacts [5]. When a synthetic static colour test pattern is used, an advanced image compressor which can optimise performance for the specific colour test pattern with the result that the metrics are not useful to compare performance with another codec. The aim of this research was to design and develop a random colour test pattern generator and gamut transfer algorithm which enables the evaluation of the colour reproduction performance of digital codecs based on any colour space and any colour gamut. A block diagram of the proposed environment is shown in Figure 1.

Most image compressors have a control parameter known as *quality factor* that can be set by the user to adjust the compression ratio. In general, the higher the compression ratio the more visible any colour artefacts become. At low compression ratios, the colour variations are not obvious to the human eye and visual appraisal is not effective. The display of these colour errors on a measurement instrument provides a better indication of the colour errors present. Since the original image is known, it is possible to determine the presence and extent of any colour artefacts.

2.1 Definition of colour components and colour space conversion

In general, it is possible to transform the red, green and blue signal values (R, G and B) to *luminance*, Y, and *chrominance* or colour difference signals, Cr and Cb, for use in image and video communication interfaces [10]. Different media applications each use

their own colour space as shown in Table 1. In general, the transformed components can be defined as:

$$Y = r_1R + g_1G + b_1B \quad (1)$$

$$C_r = r_2R + g_2G + b_2B \quad (2)$$

$$C_b = r_3R + g_3G + b_3B \quad (3)$$

where r_i , g_i , and b_i , $\{i=1, 2, 3\}$ are the coefficients of the red, green and blue signal values in a given colour space.

In analogue PAL colour television broadcasting, the two colour difference signals are used to modulate a colour sub-carrier using quadrature modulation. They can therefore be treated as two components of a vector, where the angle corresponds to the dominant colour, or *hue*, and the magnitude is the strength of the colour (or *saturation*):

$$Hue = \tan^{-1} \left[\frac{C_r}{C_b} \right] \quad (4)$$

$$Saturation = \sqrt{C_r^2 + C_b^2} \quad (5)$$

Hue, *saturation* and *luminance* defined in equations (4), (5) and (1) respectively are similar to and are compatible with analogue television measurement systems. The quantities hue and saturation as defined in equations (4) and (5) can be applied to any of the colour spaces listed in Table 1.

Many colour digital codecs use a similar approach to that used to represent colour in analogue television systems. The colour image is first transformed to luminance and chrominance. The human visual system has greater acuity to intensity than colour (chrominance); this fact allows a 2:1 chrominance compression without introducing any visually significant colour artefacts. Based on the codec type, the two chrominance components are down-sampled and coded separately. For video compression, MPEG-

2 codecs use a complex and flexible down sampling technique based on two coding parameters known as profiles and levels. However most of the colour compression standards use the luminance and colour difference signals as shown in Table 1.

In JPEG and JPEG2000, each of the two chrominance components is then coded separately using block based DCT and wavelets respectively.

2.2 The random colour test pattern

Colour bleeding is introduced by digital codecs at colour boundaries or edges. In the reconstructed image, colour bleeding appears as the blurring of the colour boundary as a result of lossy compression. Coding colour bleed is identified here as the leakage of colour from one region of colour to another at colour boundaries. Figure 2 shows an example of coding colour bleed when a digitally coded colour image having circles of six colours is reconstructed.

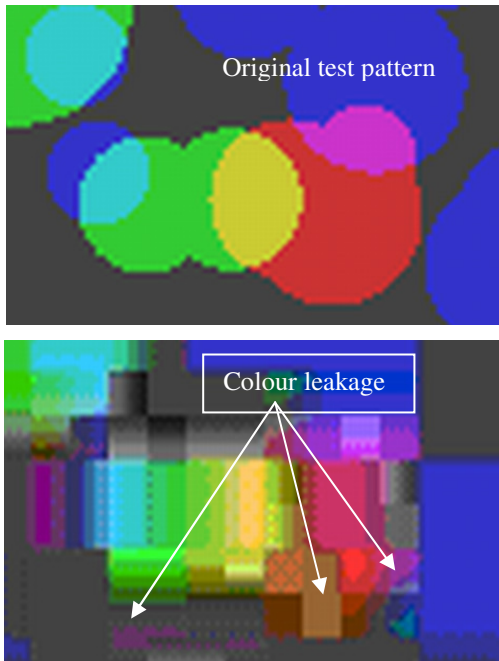


Figure 2: Example of coding colour bleed around the colour edges resulting from a JPEG codec

A random synthetic colour test pattern generator has been designed to generate a distinct test pattern each time it is run. The test pattern consists of approximately one hundred colour circles from N (in this case six) colours within a uniform background pattern of 25% grey value of full scale. The centre and the radius of the small colour circles are chosen randomly. The colour circles fit within the preset pattern size (480x640 in our tests) and they may overlap with each other. The intensity value within the circular colour regions is set to 75% of full scale. One such a random colour test pattern is shown in Figure 3. There are many forms of colour boundaries. For most colour regions, a variety of colour transitions is available so that the test pattern stresses

the codec at all compression ratios as required to emphasise the colour bleeding artefacts.

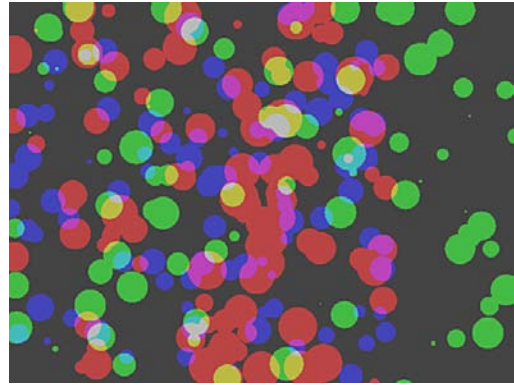


Figure 3: An original random colour test pattern designed for testing digital codecs

2.3 Definition of coding colour bleed and three artefact metrics

The *luminance*, *hue* and *saturation* colour space corresponds to the human perception system. Hence the red, green and blue components of the random colour test pattern are converted to *hue* and *saturation* prior to the calculation of artefact metrics defined in this paper.

Colour bleeding appears as a spreading of the hue angle, saturation and luminance for a colour region. The higher the leakage of colour, the higher the visibility of colour error and value of the coding colour bleed. We define coding colour bleed with three metrics, each representing the three components of colour, namely, *hue*, *saturation* and *luminance* as follows.

The human eye can not discriminate individual colour pixels in reconstructed pattern at a distance but tends to integrate over small regions. Hence the mean values of colour in individual colour regions are considered for defining metrics.

Consider a test pattern containing N distinct colours. Let the mean hue value of colour region r in the original image be \overline{H}_r and the mean hue value of the corresponding colour in the reconstructed pattern be \hat{H}_r , then the coding hue bleed can be defined as:

$$CHB \triangleq \frac{\sum_{r=1}^N \left| \overline{H}_r - \hat{H}_r \right|}{N} \quad (6)$$

Let the mean saturation value of colour region r in the original pattern be \overline{S}_r and the mean saturation value of the corresponding colour in the reconstructed pattern be \hat{S}_r , then the coding saturation bleed can be defined as:

$$CSB \triangleq \frac{\sum_{r=1}^N \left| \overline{S}_r - \hat{S}_r \right|}{N} \quad (7)$$

Let the mean luminance value of colour region r in the original pattern be \overline{L}_r and the mean luminance value of the corresponding colour in the reconstructed pattern be \hat{L}_r , then the coding luminance bleed can be defined as:

$$CLB \triangleq \frac{\sum_{r=1}^N \left| \overline{L}_r - \hat{L}_r \right|}{N} \quad (8)$$

2.4 The colour gamut

The colour gamut is defined as the subset of colours that can be represented in a given application or system. They have been defined in standards for different applications as shown in Figure 4.

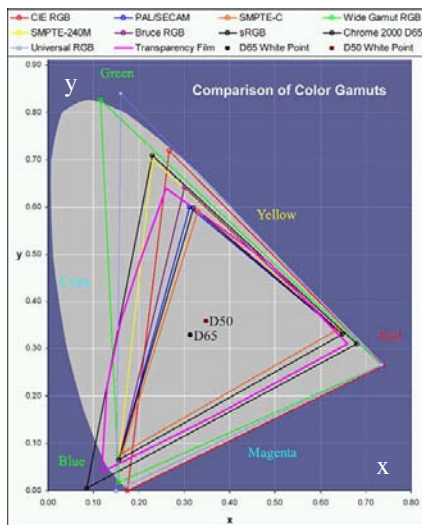


Figure 4: Comparison of colour gamut for different applications [11].

The random colour test pattern can be transformed to obtain the legal colours—the colours that can be displayed or printed within a given application. The test pattern generator creates a colour pattern based on a set of tri-stimulus values in a CIE-xy chromaticity diagram based on PAL colour gamut. The x and y values are then transformed to R , G and B values. The required colour gamut is also transformed to R , G and B values. The mapping algorithm replaces R , G and B values of the test pattern with the required values. This allows using the pattern generator for any colour gamut. Figure 5 shows a transformed test pattern of a random pattern. The algorithm maintains the structural information within the test pattern.



Figure 5: Transformed test pattern from the random colour test pattern shown in Figure 2.

3 Experiments and Results

3.1 Reconstructed test patterns

As a result of the multiplicity of edges present in the test pattern that are neither vertical nor horizontal, block processing or wavelet based compression techniques introduce errors into the reconstruction process. Figure 6 demonstrates the coding colour bleed observed when the test pattern is compressed using a JPEG codec with a compression ratio of 120. Similarly, Figure 7 demonstrates the coding colour bleed observed when the test pattern is compressed using a JPEG2000 codec with a compression ratio of 120.

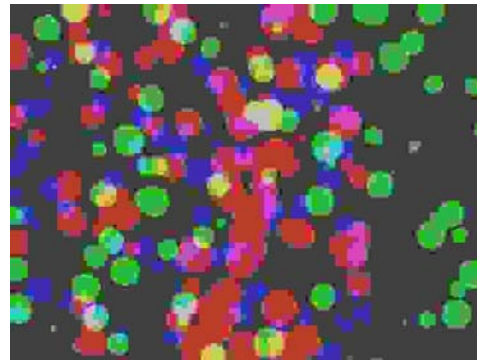


Figure 6: The reconstructed random colour test pattern when encoded with a JPEG codec with a compression ratio of 120.

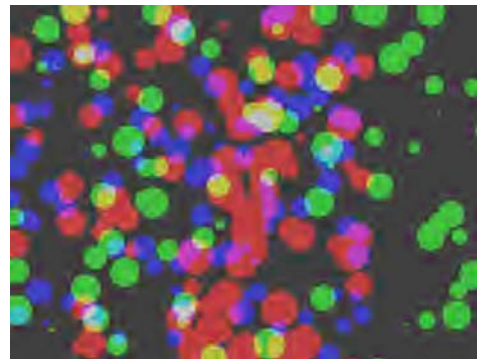


Figure 7: The reconstructed random colour test pattern when encoded with a JPEG2000 codec with a compression ratio of 120.

3.2 Three artefact metrics

The CHB , CSB and CLB artefact metrics were evaluated by applying them to the random test pattern described in the section 2.4. A JPEG and a JPEG2000 codec were tested over a range of compression ratios with the results shown in Figures 8, 9 and 10. When the random colour test pattern was compressed by a range of quality factors, this resulted in compression ratio between 2 and 230. It was observed that perceived colour errors increase with increasing compression ratio for the random test pattern with both types of codecs.

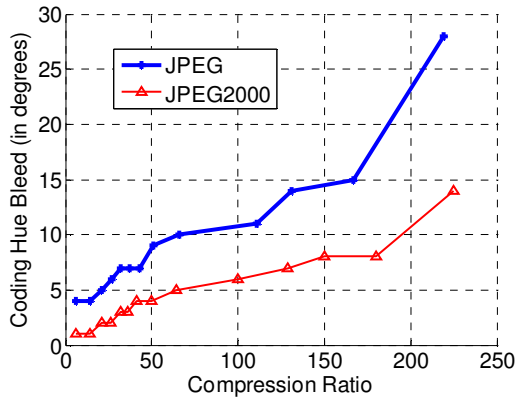


Figure 8: Coding hue bleed as a function of compression ratio with the random colour test pattern

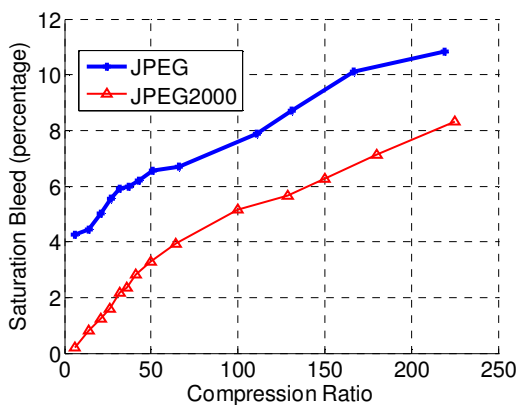


Figure 9: Coding saturation bleed as a function of compression ratio with the random colour test pattern

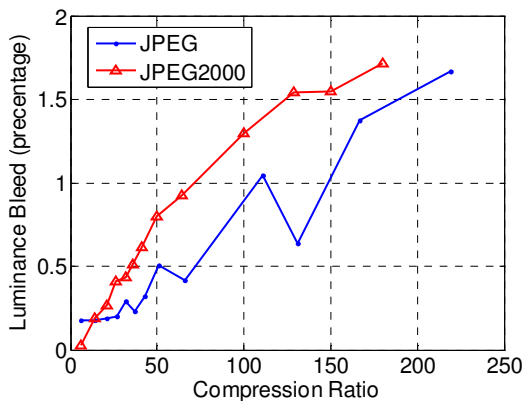


Figure 10: Coding luminance bleed as a function of compression ratio with the random colour test pattern

As shown in Figures 8, 9 and 10 the random colour test pattern also resulted in an increasing trend in all three measures of coding artefact metrics in equations (6), (7) and (8), which is in agreement with perceived quality. An increasing coding artefact metric value represents increasing bleeding artefacts. Hence the perceived quality of the reconstructed patterns decreases with an increasing bleeding measure. Coding hue bleed, coding saturation bleed and coding luminance bleed increase rapidly with increasing compression ratio. As the test pattern becomes more

compressed, the distribution of colour values becomes more spread. Minor non-monotonic variations can also be observed. At some compression levels, errors may actually reduce for increased compression depending on exactly where quantisation levels fall. The circular shape of the colour boundaries has the result that the block boundaries for JPEG will not fall on colour boundaries or parallel to them. This stresses the codec to produce more errors, which are perceivable on a monitor.

As colours from adjacent regions mix, in addition to change of hue, a significant effect of colour bleeding is a loss of saturation and luminance. This tends to make regions more grey, reducing the saturation and luminance as shown in Figure 9 and Figure 10. JPEG2000 loses slightly more intensity or luminance than JPEG codecs. The losses for both codecs are less than 2% of the original value. The human eye may not be able to make the distinction, hence can be treated as negligible.

3.3 Influence of random signal on CHB

A test was carried out to investigate the influence of random test pattern on coding hue bleed. For the same quality factor nine randomly generated test patterns were coded. As shown in Figure 11 the coding hue bleed varies with different random test patterns for the same quality factor expressed as the compression ratio. When the test pattern is coded, due to random nature, pixel contents, colour transitions and boundaries vary from one pattern to the other. As a result, for a given compression ratio, the coding colour bleed can vary by up to 20%. Hence random test pattern generator is more useful in benchmarking different codecs where each codec is fed with the same test pattern from each generation. The individual codec performance can be evaluated using the previously designed static test pattern described in [10] which enables repetitive results.

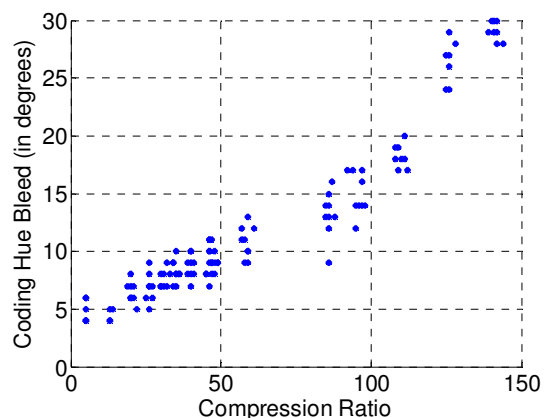


Figure 11: Variation of coding hue bleed with different random test pattern generations at different quality factors on the JPEG codec

4 Conclusions

Coding colour bleed is an undesirable visible effect found around colour edges of reconstructed, digitally coded images. In this paper, three objective artefact measures of coding colour bleed were used to evaluate and compare the colour reproduction capability of JPEG and JPEG2000 codecs with a random colour test pattern. All three colour components, namely hue, saturation and luminance are degraded in reconstructed patterns. In general, JPEG2000 performs better than JPEG in reproduction of colour despite the reduction of luminance or the intensity of the colour. Based on the random colour test pattern, it is observed that bleeding increases with increasing compression ratio. The higher the level of compression, the higher the loss of each of the colour components. The approach used is based on a known, random synthetic test pattern and measurement in each colour region of the leakage of hue, saturation and luminance made in the spatial domain. The artefact metrics provide a good representation of the coding colour bleed artefact and are readily calculated. The three artefact metrics clearly distinguish between the hue leakage, saturation leakage and luminance leakage.

The colour random test pattern proved to be useful over a wide range of compression ratios (from 2 to 240) for benchmarking two or more codecs by simultaneous testing. The random colour test pattern generator is designed with knowledge of the specific mechanisms and weaknesses inherent in compression algorithms. The JPEG image compression standard uses the discrete cosine transform (DCT) whereas JPEG2000 uses wavelets. JPEG resulted in higher colour errors compared to JPEG2000. We may deduce that wavelet based compressors would result in less colour errors compared to DCT based compressors for a given compression ratio. The gamut transfer algorithm as shown in Figure 12 enables to test codecs used in other applications by applying the proposed test pattern generator. The authors intend to perform further research to investigate the applicability of the random colour test pattern and these artefact metrics for other types of digital image and video codecs and to generalize the findings.

5 References

- [1] B. Pank, *The Digital Fact Book*. Berkshire: Quantel Limited, p. 28, 2002.
- [2] A. PUNCHIHEWA and D. G. BAILEY, "Artefacts in Image and Video Systems; Classification and Mitigation," *Proceedings of the Conference of Image and Vision Computing New Zealand*, Auckland, New Zealand, pp. 197-202, 2002.
- [3] A. PUNCHIHEWA, D. G. BAILEY and R. M. HODGSON, "Objective Quality Assessment of Coded Images: The Development of New Quality Metrics," *Proceedings of Internet, Telecommunication Conference*, Adelaide, Australia, pp. 1-6, 2004.
- [4] G. A. D. PUNCHIHEWA, D. G. BAILEY and R. M. HODGSON, "The Development of a Synthetic Colour Test Image for Subjective and Objective Quality Assessment of Digital Codecs," *The Proceedings of Asia-Pacific Communication Conference*, Perth, Australia, pp. 881-885, 2005.
- [5] A. PUNCHIHEWA, D. G. BAILEY and R. M. HODGSON, "The Development of a Novel Image Quality Metric and a Synthetic Colour Test Image for Objective Quality Assessment of Digital Codecs," *the proceedings of the IEEE Region 10 Conference (Tencon'05)*, Melbourne, Australia, 2005.
- [6] B. GROB, C. E. HERNDON, *Basic Television and Video Systems*, 6th Edition, McGraw-Hill, Singapore p. 292, 1998.
- [7] R. MAUSL, *Television Technology*, Rohde & Schwarz, pp. 9-13, 1977.
- [8] M. ROBIN and M. POULIN, *Digital Television fundamentals*, McGraw Hill, New York, USA, Second Edition, p. 126, 2000.
- [9] A. PUNCHIHEWA, D. G. BAILEY and R. M. HODGSON, "A Survey of Coded Image and Video Quality Assessment," *Proceedings of Image and Vision Computing New Zealand*, Palmerston North, New Zealand, pp. 326-331, 2003.
- [10] Rhode & Schwarz, *Sound and TV broadcasting; CCIR and FCC TV Standards*, p 3, 1977.
- [11] www.efg2.com visited 10th July 2006

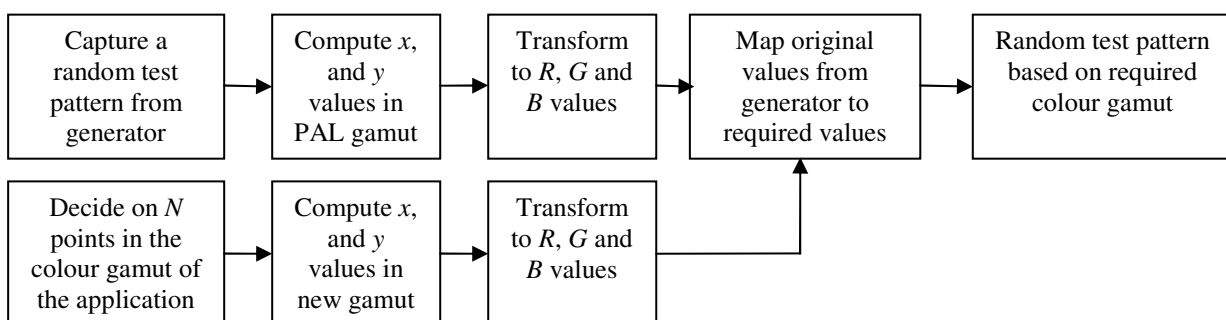


Figure 12: Block diagram of the colour gamut transformation algorithm

3D Reconstruction from an Uncalibrated Long Image Sequence

T. Osawa¹, I. Miyagawa¹, K. Wakabayashi¹, K. Arakawa², and T. Yasuno¹

¹NTT Cyber Space Laboratories, NTT Corporation, 1-1 Hikarinooka Yokosuka-Shi Kanagawa 239-0847, Japan.

²NTT Cyber Communications Laboratory Group, NTT Corporation, 1-1 Hikarinooka Yokosuka-Shi Kanagawa 239-0847, Japan.

Email: osawa.tatsuya@lab.ntt.co.jp

Abstract

In this paper, we propose a seamless and dense 3D reconstruction method that uses an uncalibrated long image sequence. First, we divide a long image sequence into subsequences. Camera motion parameters in each subsequence are estimated by a factorization method optimized for near planar motion. This is reasonable because we can assume that camera motion will be near planar motion in short sequences. Dense 3D points are reconstructed by applying a multi view stereo technique to each subsequence using the estimated camera motion parameters. We then find the relationships between the subsequences from the overlapping frames of adjacent subsequences. Finally, seamless and dense 3D reconstruction is achieved by merging all 3D points obtained from each subsequence. Experiments conducted on a long image sequence captured by a handheld camera while walking demonstrate that the proposed method yields good results.

Keywords: 3D reconstruction, Factorization, Stereo

1 Introduction

The 3D reconstruction of objects and scenes has a wide range of applications such as the digital archiving of cultural sites and modeling of environment for visual surveillance[1]. Several recently proposed methods can automatically reconstruct 3D models from short image sequences (a few hundred frames) captured by moving hand-held cameras[2, 3]. However, no study has described a method that can stably reconstruct broad-area 3D models from long image sequences.

In [4], camera motion parameters are reliably estimated by tracking several predefined markers of known 3D positions and image features. However, this approach is expensive since the measuring the marker positions needs special equipment.

We focus on extracting robust camera motion parameters by using a factorization method; the redundancy of image sequences and camera motion parameters allow us to utilize a multiview stereo technique which can reconstruct dense 3D points.

This paper proposes the combination of a factorization method and multiview stereo. The original factorization method[5] has two important problems as described below:

- Projection model is limited to orthogonal

- Feature point tracking must be performed in all frames of an image sequence

To deal with a long sequence, we first divide it into multiple subsequences and apply the factorization method optimized for planar motion. This is feasible because camera motion in a short sequence(subsequence) can be assumed to be nearly planar motion while still permitting feature points to be tracked in all frames of each image sequence. The projection model of this factorization method is perspective and camera motion parameters representing 6 degrees of freedom are reliably estimated by bundle adjustment using the planar motion obtained as the initial solution.

We apply the multiview stereo technique to each subsequence using the estimated camera motion parameters to reconstruct dense 3D points. Finally, seamless and dense 3D reconstruction is accomplished by using global coordinates to merge the 3D points computed from each subsequence.

2 Overview of the method

The proposed method can construct a 3D model from an uncalibrated long image sequence captured by a hand-held camera that was moved so as to capture the target. Figure1 overviews the proposed method.

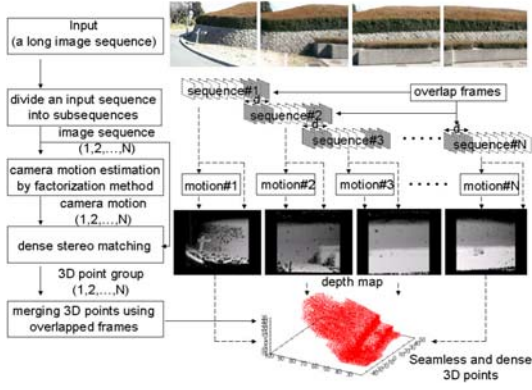


Figure 1: Overview of proposed method.

First, we divide the input, a long image sequence, into N subsequences. Each subsequence consists of F_n frames and adjacent subsequences overlap by d_n frames. F_n should be as large as possible to more robustly estimate the camera motion parameters. However, if F_n is too large, feature point tracking fails because the feature points will eventually move out of the field of view. Accordingly, F_n is determined as the maximum number of frames yielding stable feature points tracking.

The next step is to estimate the camera motion parameters by applying the factorization method optimized for planar motion to each subsequence. Camera motion parameters, camera pose $(\psi_i, \omega_i, \theta_i)$ and its 3D position $\mathbf{T}_i = (Tx_i, Ty_i, Tz_i)$, are determined in each frame. We can robustly estimate camera motion parameters (Motion# n) because camera motion closely approximates planar motion when the subsequence is short. Note that each set of camera motion parameters is given in a different local coordinate system.

At this point, camera pose and 3D position in all frames of all subsequences have been recovered. We then apply the multiview stereo technique for dense 3D point reconstruction to each subsequence. Note that each N 3D point group is also given in a different local coordinate system.

Merging the N 3D point groups is achieved by using the overlapping frames. Each overlapped frame has two sets of camera motion parameters derived from different (adjacent) subsequences. We can reconstruct the 3D points by using the two sets of camera motion parameters. Finding the relationship between the two coordinate systems is achieved by comparing the 3D points common to the two coordinate systems. We compute the relationship between all adjacent subsequences and merge the 3D point groups using a global coordinate system that can taken

as the local coordinate system of an arbitrarily selected subsequence.

3 Camera motion parameter estimation by Factorization method based on planar motion

We assume that the input, a long image sequence, is captured by a hand-held camera that is moved at walking speed so as to capture a long target. Figure 2 shows the situation envisaged and shows the coordinate relationships between the camera and 3D points. We introduce here a factorization method based on planar motion projection which is optimized for near planar motion. We derive a projection model under the assumption that the camera has near planar motion. We assume that the camera motion parameters ψ_i and ω_i contain only modest amounts of deviation. Using this assumption, we derive the approximated projection model as follows:

$$\begin{aligned} \begin{bmatrix} u_{ij} \\ v_{ij} \end{bmatrix} &= \epsilon_{ij} \begin{bmatrix} x_{ij} \\ y_{ij} \end{bmatrix} - \begin{bmatrix} \zeta_i \\ \eta_i \end{bmatrix} \\ &\approx \begin{bmatrix} \cos(\theta_i) & \sin(\theta_i) & Tu_i \\ -\sin(\theta_i) & \cos(\theta_i) & Tv_i \end{bmatrix} \begin{bmatrix} X_j/Z_j \\ Y_j/Z_j \\ 1/Z_j \end{bmatrix} \end{aligned} \quad (1)$$

where, we define ϵ_{ij} as follows.

$$\epsilon_{ij} = 1 + \omega_i \left(\frac{X_j - Tx_i}{Z_j} \right) - \psi_i \left(\frac{Y_j - Ty_i}{Z_j} \right) \quad (2)$$

$$\begin{bmatrix} \zeta_i \\ \eta_i \end{bmatrix} = \begin{bmatrix} \cos(\theta_i) & \sin(\theta_i) \\ -\sin(\theta_i) & \cos(\theta_i) \end{bmatrix} \begin{bmatrix} -\omega_i \\ \psi_i \end{bmatrix} \quad (3)$$

$$\begin{bmatrix} Tu_i \\ Tv_i \end{bmatrix} = - \begin{bmatrix} \cos(\theta_i) & \sin(\theta_i) \\ -\sin(\theta_i) & \cos(\theta_i) \end{bmatrix} \begin{bmatrix} Tx_i \\ Ty_i \end{bmatrix} \quad (4)$$

(u_{ij}, v_{ij}) is the projection point under planar motion, and (x_{ij}, y_{ij}) is the projection point observed in the image. Considering all frames $(i \in \{1, 2, \dots, F\})$ and all N 3D points, (1) can be represented as follows.

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1N} \\ u_{21} & u_{22} & \cdots & u_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ u_{F1} & u_{F2} & \cdots & u_{FN} \\ v_{11} & v_{12} & \cdots & v_{1N} \\ v_{21} & v_{22} & \cdots & v_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ v_{F1} & v_{F2} & \cdots & v_{FN} \end{bmatrix} = \begin{bmatrix} \mathbf{m}_1^T \\ \mathbf{m}_2^T \\ \vdots \\ \mathbf{m}_F^T \\ \mathbf{n}_1^T \\ \mathbf{n}_2^T \\ \vdots \\ \mathbf{n}_F^T \end{bmatrix} \begin{bmatrix} \mathbf{S}_1 \\ \mathbf{S}_2 \\ \vdots \\ \mathbf{S}_N \end{bmatrix}^T \quad (5)$$

The two vectors, $\mathbf{m}_i = (\cos(\theta_i), \sin(\theta_i), Tu_i)$ and $\mathbf{n}_i = (-\sin(\theta_i), \cos(\theta_i), Tv_i)$, represent the planar motion in the i -th frame, and vector $\mathbf{S}_j = (X_j/Z_j, Y_j/Z_j, 1/Z_j)$ is the 3D vector that represents the 3D coordinate value of the i -th point.

By assuming that the transformed points (u_{ij}, v_{ij}) are equal to the observed 2D points (x_{ij}, y_{ij}) , both the planar motion $(\mathbf{m}_i, \mathbf{n}_i)$ and the 3D coordinate value (\mathbf{S}_j) can be recovered using the factorization method based on the matrix decomposition shown in (5).

The camera motion parameters representing 6 degrees of freedom, $(\psi_i, \omega_i, \theta_i), (Tx_i, Ty_i, Tz_i)$, are estimated by bundle adjustment[6] using the planar motion $(\mathbf{m}_i, \mathbf{n}_i)$ and the 3D coordinate value (\mathbf{S}_j) as the initial solution.

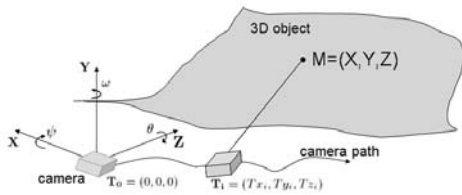


Figure 2: Relationship between camera viewpoint and target.

4 Dense 3D points reconstruction by multiview stereo

The 3D points yielded by the factorization method are very sparse because only feature points are reconstructed. However, if we know the camera motion parameters, the multiview stereo technique can be used to output a dense set of 3D points. We apply the multiview stereo technique by regarding each frame of the image sequence as a multiview image. We reconstruct the 3D points from each subsequence.

4.1 Keyframe image

Depth estimation by stereo matching using adjacent frames is unstable because the baseline in these frames is insufficient. Therefore, we select some frames as keyframes from each subsequence so that the keyframe interval provides a sufficient baseline to make depth estimation stable.

First, we select multiple base frames and decide the smallest baseline L_m in for stereo matching by using camera motion parameters that are estimated by factorization. We select multiple comparison frames for each baseframe that satisfy the condition that the baseline is longer than L_m .

After keyframe selection, we compute the projection matrix of each keyframe. The rotation matrix $\mathbf{R}_i = \mathbf{R}_{zi} \cdot \mathbf{R}_{yi} \cdot \mathbf{R}_{xi}$ and the translation matrix $\mathbf{T}_i = (Tx_i, Ty_i, Tz_i)^T$ in the i th frame are obtained from the camera motion parameters described in Section 3. The projection matrix in the i th frame is computed as follows:

$$\mathbf{P}_i = \mathbf{A} \cdot [\mathbf{R}_i, \mathbf{T}_i] \quad (6)$$

where \mathbf{A} is the camera's intrinsic parameter matrix.

4.2 Stereo matching

We employ area based matching and depth z is determined so as to minimize the SAD (Sum of Absolute Difference). To suppress mismatching in pattern-free areas, we use linear interpolation to compute the depth value z in the regions that have low $DCDX = |I_{i,j}^b - I_{i+1,j}^b|$; $I_{i,j}^b$ represents the intensity value of pixel (i, j) in the base frame.

After stereo matching all image pixels in the base frame, we compute the correspondence between the pre-rectified stereo pair using the obtained depth data. 3D points $\mathbf{M} = [X, Y, Z]$ are computed as follows:

$$\mathbf{M} = \mathbf{B}^\dagger \mathbf{b} \quad (7)$$

$$\mathbf{B} = \begin{bmatrix} up_{31} - p_{11} & up_{32} - p_{12} & up_{33} - p_{13} \\ vp_{31} - p_{21} & vp_{32} - p_{22} & vp_{33} - p_{23} \\ u'p'_{31} - p'_{11} & u'p'_{32} - p'_{12} & u'p'_{33} - p'_{13} \\ v'p'_{31} - p'_{21} & v'p'_{32} - p'_{22} & v'p'_{33} - p'_{23} \end{bmatrix}$$

$$\mathbf{b} = [p_{14} - up_{34}, p_{24} - vp_{34}, p'_{14} - u'p'_{34}, p'_{24} - v'p'_{34}]^T$$

where, (u, v) and (u', v') are corresponding pixels in the base frame and the comparison frame. p_{ij} and $p'_{i,j}$ are the (i, j) component of the projection matrix of base frame \mathbf{P} and the projection matrix of comparing frame \mathbf{P}' , respectively. \mathbf{B}^\dagger is the pseudo-inverse matrix of \mathbf{B} .

4.3 Fusing multiple depth data

We fuse the depth data computed from the base frame and multiple comparison frames by using the multiple baseline characteristic. Stereo matching using a wide baseline stereo pair is more precise in terms of depth estimation than using small baseline stereo pair, but correct matching is difficult. Setting the baseline length involves a tradeoff between precision and correctness in matching.

We assume that the depth Z_b estimated from the smallest baseline stereo pair from among multiple comparison frames is not far from the true value. Moreover, the depth Z estimated from the widest

baseline stereo pair that satisfies the condition $|Z - Z_b| < TH$ (TH :threshold value) is adopted as fused depth data. The obtained depth Z offers a lower possibility of mismatching and yields high precision because the advantages of small and wide baselines are combined.

5 Seamless reconstruction by merging 3D points

The 3D points reconstructed from each subsequence are represented using different local coordinate systems. For seamless shape reconstruction, merging all 3D points into a global coordinate system is needed. For this, we use the overlapping frames of adjacent subsequences. Each overlapping frame has two camera motion parameters derived from the two subsequences. We reconstruct the same 3D points using the two different camera motion parameters and develop a 3D point correspondence table. The conversion matrix between the two local coordinates is computed from this table. After computing the conversion matrices for all adjacent subsequences, we arbitrarily select one local coordinate system as the global coordinate system and seamless shape reconstruction is achieved by converting all 3D points into this global coordinate system.

5.1 3D point correspondence table

Creation of the correspondence table is described below. For simplicity, we consider two adjacent subsequences(subsequence# i , subsequence# $i+1$). We apply the multiview stereo technique to the overlapping frames of subsequence# i and subsequence# $i+1$ and reconstruct 3D points from depth data using projection matrix \mathbf{P}^i computed from subsequence# i and projection matrix \mathbf{P}^{i+1} computed from subsequence# $i+1$. All pixels (u, v) of the base frames have two 3D points, one is computed using \mathbf{P}^i and the other is computed using \mathbf{P}^{i+1} (Figure3). These two 3D points are the same point, but represent the use of two different coordinate systems. Thus, we can obtain an enormous number of 3D point correspondences so the estimation of the conversion matrix is extremely stable.

5.2 Merging 3D points

We find the relationship(scale factors, translation (Tx, Ty, Tz) and rotation ψ, ω, θ) between the local coordinates derived from adjacent subsequences as a conversion matrix using the 3D point correspondence table. To compute the conversion matrix, we employ the method[7] that

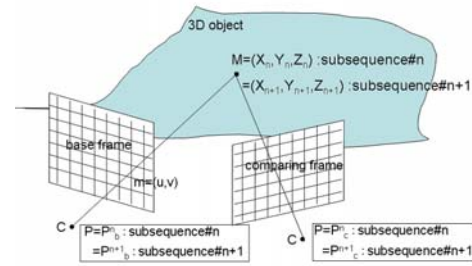


Figure 3: 3D point correspondence.

uses unit quaternion. We take a local coordinate system derived from the center subsequence in the time-series as the global coordinate system. To merge all 3D points into this global coordinate system, we sequentially convert 3D points between adjacent subsequences. This results in a seamless and dense 3D model.

6 Experiments

6.1 Flow of experiments

In this experiment, we use a hand-held digital camera(DimageZ3) made by KONICA MINOLTA. This hand-held camera can capture VGA image sequences at 30fps. We captured long sequences while walking along the target. Figure4 shows a typical sequence. The significant level of scene flow makes it impossible to track feature points in all frames. First, we divided each long sequence



Figure 4: Typical long image sequence

into subsequences. The number of frames of subsequence# n F_n is defined as feature points detected in center frame in subsequence are not frameout and can be tracked stably. In these experiments, we consider walking speed while capturing is nearly constant, so we set F_n and d_n which is number of overlapping frames are fixed number.

We detected feature points in the center frame of each subsequence and tracked these feature points forward and backward in the time series. Camera motion parameters were estimated by using the result of feature points tracking. Multiview stereo was applied for dense 3D points reconstruction in each subsequence. Reconstructed 3D points from

each subsequence were merged into a global coordinate system. Fig5 shows an example of reconstruction.



Figure 5: Example of integrated 3D points

6.2 Evaluation experiment

To confirm the performance of the proposed method, we conducted a evaluation experiment. We evenly spaced 5 room dividers and captured a long sequence by a hand-held camera while walking in front of the dividers. Fig6 shows the experimental setup.

The captured long sequence consisted of 300 frames. We divided this sesquence into 7 subsequences and reconstructed the 3D points by our proposed method.

Fig7 shows a bird's eye view of the reconstruction result. To evaluate the deviation from flatness of the reconstructed dividers and reconstruction errors, we computed the standard deviation from flatness, divider spacing intaevals and relative angle of dividers(relative to divider1).

Table1 shows the results of this evaluation. Our proposed method yielded very good results, considering that the capture distance from each divider was about $2m \sim 4m$.

With regard to standard deviation, the center dividers had worse reconstruction accuracy than divider1 or divider2.

Fig8 shows the relationship between distance and number of 3D points. Reconstructed center dividers exhibited a bimodal distribution while the end dividers exhibited a unimodal distribution. This difference was due to merging errors, because the center dividers appeared in many more subsequences than the endmost dividers.

Note that the standard deviation of divider3, the worst case, was only 1.79cm, which shows that the proposed method achieved very stable merging of 3D points.

6.3 Experiment in a real environment

To confirm the proposal's performance in a real environment, we applied it to the long sequence shown in Fig4.

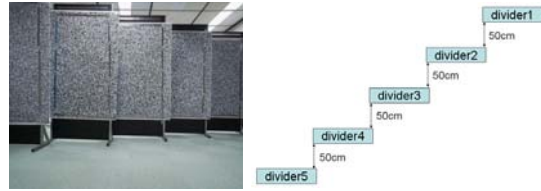


Figure 6: Experimental setup

Table 1: Evaluation results

	SD(cm)	Angle(degree)	Distance(cm)
divider1	1.11	0.00	0.00
divider2	1.44	2.15	47.32
divider3	1.79	1.44	95.75
divider4	1.45	2.94	147.38
divider5	1.12	5.75	202.47

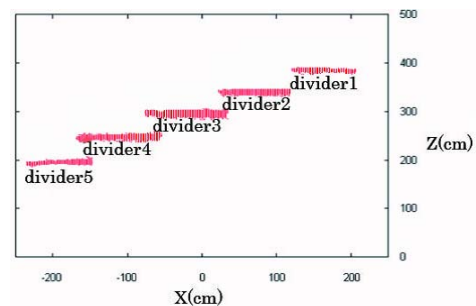


Figure 7: Reconstructed divider group

This long sequence consisted of 1100 frames. Fig5 shows the reconstruction of the dence 3D points. For visualization, we colored each 3D points using the color of corrsponding image pixel. Final result(Fig9) shows that the proposed method reconstructed seamless and dence 3D shape in a real environment.

7 Conclusions and future works

This paper proposed an automatic 3D reconstruction method that takes as input long image sequences captured by hand-held cameras. The method divides each long sequence into short subsequences and merges the 3D points that are reconstructed from each subsequence into a global coordinate system.

Experiments confrimed the good performance of our method with regard to deviation from flatness, angle, and interval of reconstructed room dividers.

In future work, we will tackle more precise 3D reconstruction by re-estimating camera motion in a global coordinate system and reconstruct 3D points using all frames of a long image sequence.

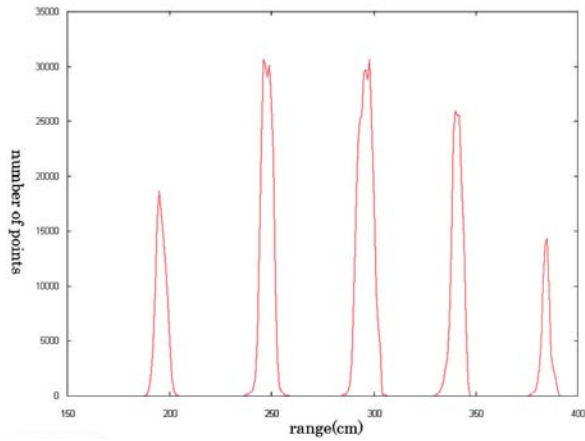


Figure 8: Relationship between number of 3D points and range

References

- [1] T. Osawa, X. Wu, K. Wakabayashi, and T. Yasuno, "Human tracking by particle filtering using full 3d model of both target and environment," in *Proc. IAPR International Conference on Pattern Recognition.*, 2006.
- [2] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch, "Visual modeling with a hand-held camera," *International Journal of Computer Vision*, vol. 59, no. 3, pp. 207–232, 2004.
- [3] M. Lhuillier and L. Quan, "A quasi-dense approach to surface reconstruction from uncalibrated images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 418–433, 2005.
- [4] T. Sato, M. Kanbara, N. Yokoya, and H. Take-mura, "Dense 3-d reconstruction of an outdoor scene by hundreds-baseline stereo using a hand-held video camera," *International Journal of Computer Vision*, vol. 47, no. 1–3, pp. 119–129, 2002.
- [5] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: A factorization method," *International Journal of Computer Vision*, vol. 9, no. 2, pp. 137–154, 1992.
- [6] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment a modern synthesis," in *Proc. Int. Workshop on Vision Algorithms*, pp. 298–372, 1999.
- [7] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America A*, vol. 4, pp. 629–642, 1987.

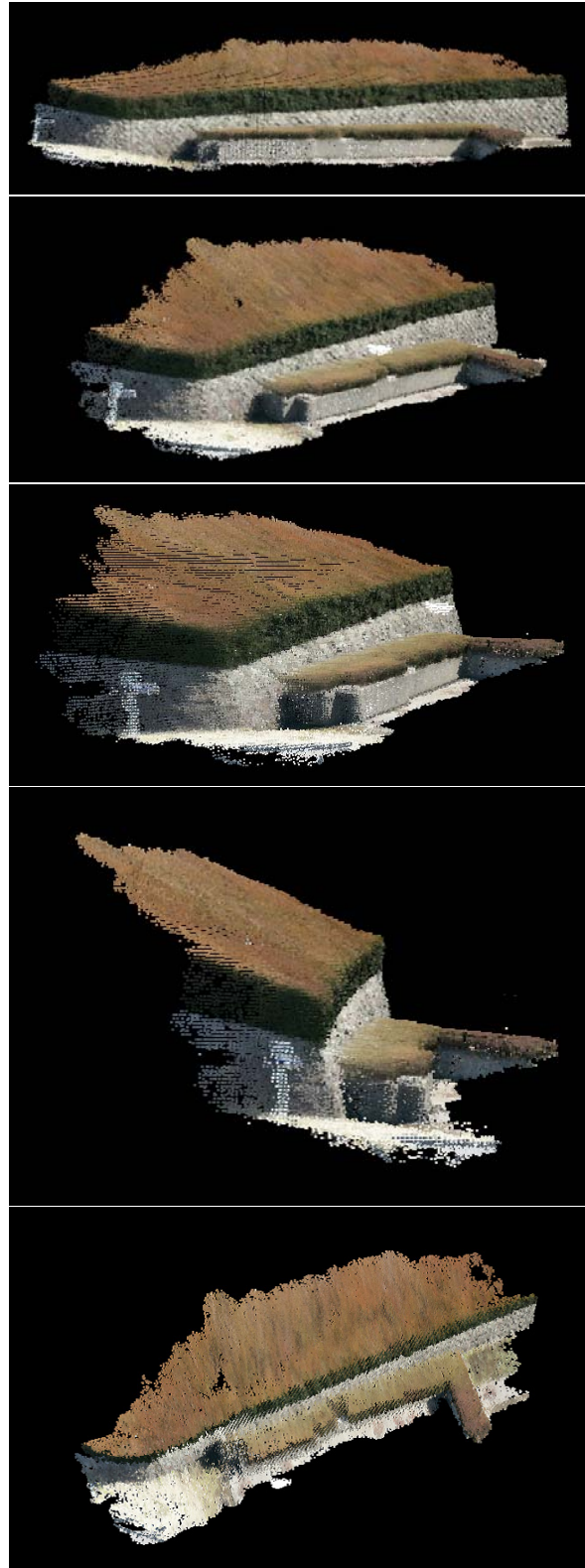


Figure 9: Reconstructed 3D shape model

Stereo Vision: Concurrent Matching *vs.* Optimisation

Georgy Gimel'farb, John Morris, Patrice Delmas, Jiang Liu

Department of Computer Science, Tamaki Campus

The University of Auckland, Private Bag 92019

Auckland 1000, New Zealand

Email: {g.gimelfarb,j.morris,p.delmas}@auckland.ac.nz

Abstract

As an ill-posed inverse problem, stereo vision has a multiplicity of equivalent solutions. Conventional best matching of stereo images under regularising constraints is an NP-hard optimisation problem with only approximate solutions. Our alternative, computationally feasible framework, concurrent stereo matching, finds instead a set of all candidate 3-D points and use them to constrain the desired solution. The experiments reported here firstly demonstrate the magnitude of the noise problem for stereo matching - in the Middlebury set, $> 50\%$ of pixels do not match to the corresponding pixel in the other image with $< 1\%$ discrepancy. When the contrast range of one image is altered, CSM consistently produces fewer errors than graph-cut or belief-propagation algorithms. CSM is also robust to image corruption with additional Gaussian noise, performing better than GC, BP or SDPS algorithms.

Keywords: Stereovision, stereo correspondence, CSM

1 Introduction

Most existing solutions of the stereo vision problem use a similar optimisation framework: find a *single* surface minimising signal dissimilarity between corresponding points in a stereo pair under constraints which identify a solution for an inherently ill-posed inverse optical problem. Finding the optimal solution is NP-hard, but in some cases there exist computationally feasible approximate solutions which provably reach a vicinity of the optimum [2, 4]. The signal dissimilarity is measured under simplifying constraints (i.e. ordering or equal noiseless corresponding signals) to make the problem at least approximately solvable. However most of these assumptions are weak in practice. The ordering constraint is invalid when an observed scene contains disjoint optical surfaces, e.g. the foreground poles in Fig. 1. Real optical surfaces are not Lambertian, so the corresponding image points have unequal intensities and the scene may contain both opaque and semi-transparent surfaces, e.g. the elevator tower in Fig. 1.

For many years, professional stereophotogrammetry exploits only least squares correlation based matching scores because slowly varying contrast and offset deviations are typical for real stereo pairs. However, even the best-performing computer vision algorithms (e.g. those ranked on the Middlebury test site [10, 11]) fail when even relatively small uniform contrast and offset



Figure 1: Rectified stereo pair 'Building'. This scene was captured by the same camera at different times: note differences in clouds, palms, and pedestrians as well as semi-transparent surfaces.

transformations are applied to one image of a stereo pair to mimic real conditions: rank on this popular test bench is based on specific stereo pairs with no contrast or offset deviations. Most conventional optimisation approaches such as graph-cut (GC) [1, 2] or belief propagation (BP) [3, 12] techniques cannot account for such deviations and thus show large errors on more realistic stereo pairs [6] (see also section 4.3).

Because optimisation frameworks select an approximate solution from the multiplicity of equivalent ones (that are possible because the problem is ill-posed) and are typically based on unrealistic noise models and simplifying assumptions, we have advocated an alternative approach, called concurrent stereo matching (CSM) [8, 9] that abandons the search for a single optimal optical surface providing the least dissimilarity between the two images.

Both stereo images are noisy, therefore the closest pixel-to-pixel image matches need not represent actual correspondences. However, if we generate a set of all ‘good’ or ‘close’ matches under a realistic noise model then the actual correspondences should not be lost. CSM has three sequential steps:

- estimate image noise (including contrast and offset deviations and possible partial occlusions) in a stereo pair
- select candidate volumes using signals matching within noise model criteria and
- fit smooth opaque surfaces to the volumes with due account of weak visibility constraints (i.e. that each pixel is produced by a single surface point).

The paper is organised as follows. Some examples of the effects of noise are presented in Section 2 to illustrate the magnitude of the problems introduced by noise in image matching. Section 3 presents the symmetric model of stereo images and digital optical surfaces on which our approach is based. Section 4.2 presents a new matching score to select candidate volumes which allows for contrast and offset deviations, random noise and outliers (mismatches). Section 4.2.1 presents results of empirical evaluation of true and estimated image noise. Section 4.3 discusses the CSM framework and presents experimental results.

2 Noise in Stereo Pairs

Here, we show the magnitude of ‘noise’ found in typical stereo pairs, using the Middlebury benchmarks: the ground truth determines which pixels *should* match in the two images. The distribution of intensity differences for supposedly matching pixels is shown in Table 1. Let

$$\delta_{LR}(x, y, d) = |g_L(x + \frac{d}{2}, y) - g_R(x - \frac{d}{2}, y)|$$

$$\delta(x, y) = \min_{d \in [0.. \Delta]} \delta_{LR}(x, y, d)$$

and

$$\delta_{gt}(x, y) = \delta_{LR}(x, y, d_{gt}(x, y))$$

where $d_{gt}(x, y)$ is the value of the disparity at (x, y) obtained from the ground truth and $g_{L|R}$ are grey-level intensities, defined formally in Section 3.1. In all cases, the number of *exactly matching* ($\delta(x, y) = 0$) points is less than 20% of the total and allowing one unit of intensity difference raises this to less than 50%. In Table 1, f_m is the number of points for which $\delta_{gt}(x, y, d) \in [\delta_a, \delta_b]$. ($\delta_{gt} = 0$ represents a ‘perfect’ match.) f_a shows the number of points for which the intensity difference,

$\delta_{LR}(x, y) \in [\delta(x, y) + \delta_a, \delta(x, y) + \delta_b]$. In the column headed ‘0’, $f_a - f_m$ is the number of points for which the ‘best’ match is not necessarily the correct one! The second column shows the number of points for which a another match is possible within 1 intensity value (from a range of 0-255), *etc.* The multiplicity of sources for ‘noise’ in stereo pairs listed in Table 2 is responsible for this and makes inclusion of realistic noise models imperative if correct scene geometry is to be recovered. In CSM, we avoid simplistic assumptions (*e.g.* zero means) and allow more sophisticated noise models to be incorporated into the scene recovery process.

Table 1: Percent of matching points within the indicated intensity difference range, $\delta_a.. \delta_b$ (see text)

δ_a	0	1	2	6	11	21	61
δ_b	0	1	5	10	20	60	255
Tsukuba							
f_m	18.9	30.3	39.3	6.5	3.4	1.5	0.1
f_a	30.4	37.5	26.2	3.1	1.7	1.0	0.1
Venus							
f_m	18.3	29.7	34.4	8.4	5.6	3.1	0.5
f_a	28.4	36.5	25.2	5.0	3.1	1.7	0.1
Teddy							
f_m	11.4	21.2	44.7	10.1	5.4	3.8	3.4
f_a	14.1	23.8	36.8	8.2	6.3	6.2	4.6
Cones							
f_m	5.2	10.5	43.2	23.5	8.5	7.0	2.1
f_a	7.9	14.9	39.0	16.0	10.1	10.2	1.9

3 Symmetric stereo image model

For simplicity, we consider a canonical symmetric camera geometry - identical cameras with parallel optical axes and image planes and collinear scan lines (x -directed conjugate scan lines have the same y coordinates). We work in the space of a virtual Cyclopæan image, which is aligned with the left and right images and has an origin on the baseline half-way between the real camera optical centres. In this Cyclopæan space, x and y coordinates map directly to x and y coordinates in the scene space but an orthogonal third axis represents x -disparities, d , of the corresponding points, i.e. it is the reciprocal of the scene depth or z axis distance. Epipolar lines follow scan lines in the x direction in this configuration, so that matching always examines points with common y coordinates. A graph, $G = [N; E]$, with nodes (representing scene points imaged onto the centres of individual pixels) $N = \{(x, y, d) : x = 0, \dots, M - 1; y = 0, \dots, N - 1; d = d_{\min}, \dots, d_{\max}\}$ and edges, E , between the nodes can describe all

Table 2: Stereo ‘Noise’ Sources

Source	Noise types
Signal	electromagnetic interference (<i>e.g.</i> cross-talk) quantum behaviour of electronic devices (<i>e.g.</i> resistor shot-noise) quantization noise from digitizing real-valued analogue signals
Electronics	intensity sensitivity variations between cameras (<i>e.g.</i> different optical or electronic gain settings), different ‘dark noise’ levels
Geometry	Discrete pixel sensors with finite area, Occlusions, Perspective distortion
Optics	different camera intrinsic parameters, <i>e.g.</i> focal length, lens distortion, <i>etc.</i> non-uniform scattering (non-Lambertian surfaces), blurring due to finite depth-of-field of lens systems, reflections and specular highlights, angle dependent colour scattering (‘grating’ effects), lighting variation due to different view angles

observable variants of optical surfaces - quantized by the discrete pixel nature of the sensor.

An observed scene may contain multiple surfaces, but every camera observes a single, but possibly disjoint surface. Conventional stereo algorithms assume a single continuous opaque surface having - at most - self-occlusions but no folds along the d -axis. In the Cyclopæan image, a node (scene point) has five visibility states:

- T - transparent,
- B - binocularly visible opaque surface point,
- ML (or MR) - a monocularly visible opaque point observed only by the left (or right) camera and occluded by the same surface from the other camera
- I - invisible occluded by the surface from both cameras.

The states T and I are implicitly specified by any surface variant and need not appear explicitly, thus the surface model may contain points in states: B, ML or MR [5] - see Fig. 2. Due to the symmetry of the Cyclopæan viewpoint and visibility constraints, edges from every node (x, y, d) on a surface only link the three nearest nodes, $\{(x + 0.5, y, d - 1); (x + 1, y, d); (x + 0.5, y, d + 1)\}$.

More realistic 3-D scenes with multiple disjoint surfaces have a more flexible viewing geometry. Generally, several opaque surfaces exist in the d -direction for every (x, y) -point in the Cyclopæan space, although every camera observes only a single but possibly discontinuous surface formed from the visible parts of all the surfaces. The transparent

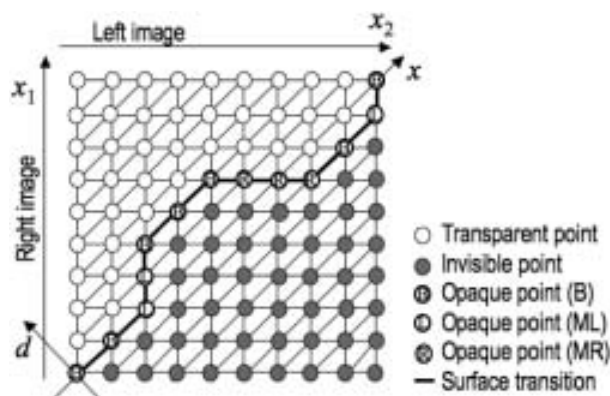


Figure 2: Single continuous (x, d) -profile of an optical surface in the cyclopæan (x, y, d) -space.

points also have three states depending on discontinuities in the scene, namely, fully transparent points (T), points transparent only for the left (TL) or right (TR) camera but occluded from the other camera as shown in Figure 3. In general, the ordering constraint does not hold and surface discontinuities may or not impact transparency of the intermediate points.

3.1 Signal model - including noise

Let $\mathbf{Q} = \{0, \dots, Q - 1\}$ denote a finite set of grey levels (intensities) and $\mathbf{R} = \{(x, y) : x = 0, \dots, M - 1; y = 0, \dots, N - 1\}$ be a finite arithmetic lattice supporting digital images. Formally, we assume that the Cyclopæan imageviewed from the midpoint of the baseline - $g_c : \mathbf{R} \rightarrow \mathbf{Q}$ - is the reference (true) image and consider the left and right images - $g_L : \mathbf{R} \rightarrow \mathbf{Q}$

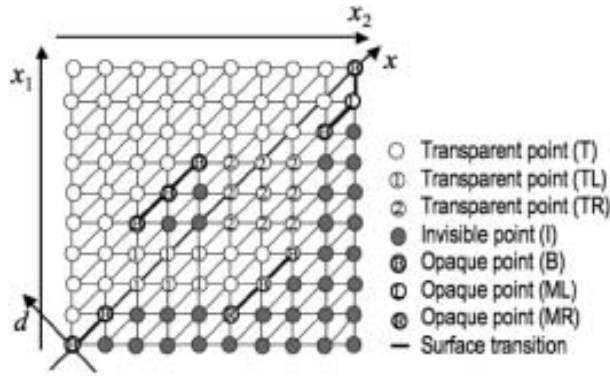


Figure 3: Example of multiple disjoint (x, d) -profiles in the Cyclopæan (x, y, d) -space.

and $g_R : \mathbf{R} \rightarrow \mathbf{Q}$ - derived from it¹. Let $n : \mathbf{R} \rightarrow \{-Q + 1, \dots, 0, 1, \dots, Q - 1\}$ denote a noise component of an intensity in a real image. The left (L) and right (R) images may have different gain (or contrast) values, $\alpha_{L|R}$, offsets, $\beta_{L|R}$ and noise contributions, $n_{L|R}$:

$$\begin{aligned} g_L(x + \frac{d}{2}, y) &= \alpha_L g_c(x, y) + \beta_L + n_L(x, y) \\ g_R(x - \frac{d}{2}, y) &= \alpha_R g_c(x, y) + \beta_R + n_R(x, y) \end{aligned} \quad (1)$$

Note that it is possible to assume that the Cyclopæan image has the same contrast and offset parameters as one of the images, *e.g.* the left one, and set $\alpha_L = 1, \beta_L = 0$, reducing the number of adjustable parameters. The simplest noise model has two components:

1. a centred Gaussian or more general symmetric noise with zero mean $\mathbb{E}\{n(x, y)\} = 0$ over corresponding areas in the images, and
2. outliers uniformly distributed over the range of permissible intensities in an image.

Generally, both the distributions should approximate real empirical distributions of the candidate matches and mismatches over the images [7] (*cf.* Section 4.2.1).

The basic idea underlying the model (Eq. 1) is that signal differences vary slowly across the lattice but remain relatively small in the matched areas if their global contrast, α , and offset, β , deviations are excluded, whereas there will be outliers with large differences and arbitrary changes. Thus, simple signal differences, $\delta_{LR}(x, y, d)$, used almost universally in matching scores simply do not work.

¹Note that this model assigns sequences of monocularly visible points to disparities lying on straight lines between binocularly visible points, see Figure 2. This is a reasonable assumption - and, in the absence of additional, *e.g.* context, information, no better assumption can be made for a monocularly visible point.

4 Concurrent Stereo Matching

4.1 Step 1 - Estimating the noise

Ideally, we should be able to determine some parts of the noise from camera calibration, scene and lighting measurements, but - as with the benchmark images commonly used for assessing stereo algorithm performance[11] - this is often not feasible and it is necessary to determine noise from the images alone.

Here we address several aspects of ‘noise’ explicitly rather than try to bundle all sources into an approximate model. Firstly, we allow for spatially variant noise and do not fit a global model - implicit in conventional correlation algorithms².

Initially, the image is segmented over each line into regions which show low variance in signal intensity. Regions of similar *colour* (using the H component in an HSI colour space) in the other image are then sought and the signal contrast and offset deviations as well as the random signal noise are sought by solving Eq. (1) over the region.

4.1.1 Pixelization errors

Object boundaries are distinguished by significant changes in optical signal³. This produces an ‘un-matchable’ pixel if the disparity is not close to an integral value - the boundary in one image will fall in the middle of one pixel, which acquires an intensity between the actual intensities of both objects.

4.2 Matching with outliers

At each disparity level, d , let a soft label or weight, $\gamma_{x,y} \in [0, 1]$, indicate the probability that a pixel is a candidate match and not an outlier. This probability is inversely monotone with respect to the absolute deviation in order to allow the noise to have the same standard deviation, σ , in each image. Then the symmetric least squares matching score $\Phi_{g_L;g_R;d}$ is obtained by the maximum likelihood estimation of the noise parameters $\theta = \{\alpha_L, \alpha_R, \beta_L, \beta_R, g_c\}$ - outlier model of Eq. (1):

$$\Phi_{g_L;g_R;d} = \min_{\theta} \Phi_{g_L;g_R;d}(\theta) \quad (2)$$

where

$$\Phi_{g_L;g_R;d}(\theta) = \sum_{(x,y) \in \mathbf{R}} \gamma_{x,y} (n_L^2(x, y) + n_R^2(x, y)),$$

²Adaptive window algorithms may claim to include spatially variant noise factors, but the approach is simplistic - it simply varies the (usually rectangular) region over which effective averaging takes place.

³The pathological case of two overlapping textureless objects of similar colour defeats any matching process and is not considered further.

$n_L^2(x, y) = (g_L(x + \frac{d}{2}, y) - \alpha_L g_c(x, y) - \beta_L)^2$ and $n_R^2(x, y) = (g_R(x - \frac{d}{2}, y) - \alpha_R g_c(x, y) - \beta_R)^2$. The score was obtained by an EM-based local minimisation that re-evaluated the weights, $\gamma_{x,y}$, in every iteration. After the current score is found using the previous weights, the next weights follow from a simple rule describing how the two main types of noise contribute to every signal difference:

$$\gamma_{x,y} = \frac{\kappa p_{\text{match}}(n_L(x, y), n_R(x, y))}{\kappa p_{\text{match}}(n_L(x, y), n_R(x, y)) + (1 - \kappa) p_{\text{outlier}}}$$

Here, κ is a prior probability for candidate matches and $p_{\text{match}}(\dots)$ and p_{outlier} are current joint probability densities of the noise values for the pixel pair considered as the candidate match and the outliers, *e.g.* the joint Gaussian density where the variance relates to the matching score and the uniform density, respectively:

$$p_{\text{match}}(n_L(x, y), n_R(x, y)) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{\nu_L^2 + \nu_R^2}{2\sigma^2}\right);$$

$$p_{\text{outlier}} = \frac{1}{\nu_{\text{max}}^2}$$

By setting the derivatives of the unknown parameters to zero and some algebra, we obtain

$$\Phi_{g_L;g_R;d} = \frac{1}{2} \left(S_{LL} + S_{RR} - \sqrt{(S_{LL} - S_{RR})^2 + 4S_{LR}^2} \right)$$

the estimated noise variance is $\sigma^2 = \frac{\Phi_{g_L;g_R;d}}{\Gamma}$, and the joint squared residual noise for each pixel is:

$$n_L^2(x, y) + n_R^2(x, y) = (a_R \tilde{g}_L(x, y) - a_L \tilde{g}_R(x, y))^2$$

where

$$a_L^2 = \frac{1}{2} \left(1 + \frac{S_{LL} - S_{RR}}{[(S_{LL} - S_{RR})^2 + 4S_{LR}^2]^{\frac{1}{2}}} \right)$$

$$a_R^2 = \frac{1}{2} \left(1 - \frac{S_{LL} - S_{RR}}{[(S_{LL} - S_{RR})^2 + 4S_{LR}^2]^{\frac{1}{2}}} \right)$$

$$S_{ij} = \sum_{(x,y) \in \mathbf{R}} \gamma_{x,y} \tilde{g}_{i;x,y} \tilde{g}_{j;x,y}; \quad i, j \in \{L, R\}$$

$$\tilde{g}_{L;x,y} = g_L(x + \frac{d}{2}, y) - \frac{1}{\Gamma} \sum_{(x,y) \in \mathbf{R}} \gamma_{x,y} g_L(x + \frac{d}{2}, y)$$

$$\tilde{g}_{R;x,y} = g_R(x - \frac{d}{2}, y) - \frac{1}{\Gamma} \sum_{(x,y) \in \mathbf{R}} \gamma_{x,y} g_R(x - \frac{d}{2}, y)$$

$$\Gamma = \sum_{(x,y) \in \mathbf{R}} \gamma_{x,y}$$

The weights $\gamma_{x,y}$ are re-evaluated at every iteration. The iterations terminate when $\Phi_{g_L;g_R;d}$ becomes almost constant. The weights indicate the candidate matches such that $\gamma_{x,y} \geq \gamma_0$ where γ_0 is a threshold in the range $[0.5, 1]$. These matches at different disparity levels for the images in Fig. 1 are shown in Fig. 4.

4.2.1 Empirical noise distributions

Signal mismatches for points which are expected to match, based on the ground truth, are shown in Table 1. Empirical distributions of the absolute residual joint noise estimate, $\delta = \sqrt{n_L^2(x, y) + n_R^2(x, y)}$, are shown in Table 3: f_a is the percentage of points with noise in the indicated range for all disparity levels, whereas f_i is the percentage of points with minimum noise.

Table 3: Distribution of residual noise (f_a – percent of points (x, y, d) with noise δ in the indicated range for all disparity levels; f_i – percent of point with minimum noise in the indicated range).

δ	0-0	1-1	2-5	6-10	11-20	21-60	61-255
Tsukuba							
f_a	24.3	11.8	25.5	10.7	10.7	14.5	2.5
f_i	93.5	2.7	3.8	0.0	0.0	0.0	0.0
Venus							
f_a	26.7	15.3	23.5	9.6	9.2	11.6	4.1
f_i	95.6	2.1	2.3	0.0	0.0	0.0	0.0
Teddy							
f_a	9.5	7.7	22.1	13.3	14.7	22.6	10.1
f_i	95.8	2.2	2.0	0.0	0.0	0.0	0.0
Cones							
f_a	5.6	5.4	16.2	14.0	18.7	30.6	9.5
f_i	91.5	5.1	3.4	0.0	0.0	0.0	0.0
Building							
f_a	4.3	3.8	13.2	11.5	14.9	28.1	24.2
f_i	87.4	4.2	5.6	1.7	1.1	0.0	0.0

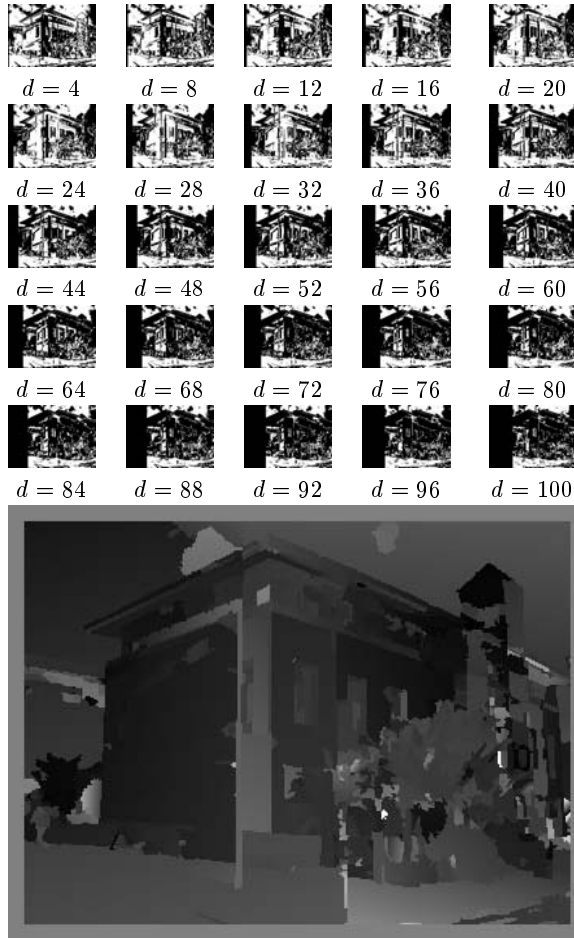
4.3 Step 3 - Surface fitting

Once noise has been estimated for each pixel in the image, selection of candidate volumes commences. For the high resolution, large disparity range (0..100) image pair, ‘Building’ (Figure 1), part of the set of candidate volumes for the are shown in white in Figure 4 (upper section). Then the surface fitting phase is entered and volumes are selected from the candidates using procedures previously described [8, 9, 6] with the final result shown in Figure 4 (lower section).

Note that the poles in the foreground (which violate ordering constraint often implicitly used by matching algorithms) are substantially correctly identified. By retaining all candidate matches until the third (surface-fitting) phase, CSM successfully reconstructs the disjoint surfaces behind the poles.

5 Experiments

To demonstrate the ability of CSM to process images with different contrast and offset settings,



Frontal-planar disparity map

Figure 4: Upper: Candidate matches (white) at different disparity levels $d \in [0..100]$ for the stereo pair in Fig. 1; Lower: frontal disparity planes that best fit the candidate volumes.

we perturbed the left image of the pair by altering the intensity range (contrast) by various amounts [6]. In the results presented in Table 4, a contrast change of $x\%$ means that the range - minimum intensity to maximum intensity - was reduced ($x < 0$) or expanded ($x > 0$) by the indicated value. Saturation arithmetic was used.

The results show that NCSM lags in performance (sometimes by only small margins) relative to graph-cut and belief-propagation algorithms for undistorted (0%) images. However, it produces consistently better results both when the contrast was increased and decreased. Symmetric dynamic programming also shows good tolerance to contrast variations: NCSM performs generally slightly better but the margin is sometimes quite small and within experimental error: some experiments show SDPS performing better.

Sensitivity to added Gaussian noise was also tested: noise of several power levels - measured

Table 4: Mean reconstruction errors after contrast variations for graph-cut (GC) [14], belief propagation (BP) [13], SDPS [5] and NCSM [8, 9, 6] algorithms

	% change in left image contrast						
	-60	-40	-20	0	+20	+40	+60
Tsukuba [10, 11]							
GC	7.1	7.0	6.1	1.0	6.6	7.2	7.9
BP	6.9	6.1	5.3	1.3	5.5	6.4	7.2
SDPS	3.1	1.8	1.1	1.2	1.3	1.9	3.0
NCSM	1.4	1.2	1.2	1.2	1.4	2.8	3.6
Venus [10, 11]							
GC	7.1	7.0	6.3	0.8	6.6	7.2	7.6
BP	5.8	5.5	4.7	0.7	6.7	7.7	8.4
SDPS	3.2	1.3	1.1	1.0	1.0	1.5	3.0
NCSM	1.7	1.7	1.7	1.2	1.2	1.9	2.5
Cones [10, 11]							
GC	6.8	6.0	4.6	3.1	4.6	7.9	8.1
BP	8.2	7.1	5.0	3.6	7.5	9.5	9.9
SDPS	5.3	4.1	3.0	3.0	3.0	4.2	5.0
NCSM	3.5	3.4	3.4	3.5	3.5	3.4	4.7
Teddy [10, 11]							
GC	7.1	7.0	6.1	3.1	6.5	7.3	7.8
BP	8.0	7.7	5.5	3.9	7.8	10.0	11.1
SDPS	6.5	5.1	4.5	3.0	4.0	4.0	6.9
NCSM	4.3	3.2	3.5	3.1	3.0	3.8	5.1

by the variance, σ^2 - was added to the images. Reconstruction errors are set out in Table 5. Again, although it does not perform as well as the GC or BP algorithms at the low levels of noise present in the somewhat ideal images in the Middlebury set, NCSM outperforms the others with increasing noise. It presents better results for all experiments with $\sigma \geq 20$, bar one, in which GC achieves 5.5% errors compared to NCSM's 6.0%.

6 Conclusion

(i) Matching is not minimal - for every true match (match at the ground truth disparity), similar (and often 'better') matches exist at other disparities. Thus simplistic minimum of signal differences approaches lead to solutions deviating from the ground truth and extra constraints on the surfaces must be invoked to approach a solution.

(ii) Concurrent matching separates the choice of matches from the surface fitting. In the matching phase, all reasonable matches are accepted. This creates a more natural noise-driven process: match images using a noise model that takes account of contrast and offset deviations and occlusions (outliers). This endows our matching algorithm with an ability that human brains show without effort.

Table 5: Reconstruction errors with added Gaussian noise

σ	0	10	20	30	40	50	60
Tsukuba [10, 11]							
GC	1.0	1.5	2.3	3.0	3.1	3.6	3.9
BP	1.3	1.6	2.5	3.2	3.5	3.8	4.1
SDPS	1.2	1.4	1.8	2.3	2.6	2.8	2.9
NCSM	1.2	1.2	1.3	1.6	1.9	2.3	2.5
Venus [10, 11]							
GC	0.8	1.8	2.7	4.7	5.4	6.0	6.1
BP	0.7	1.8	4.0	4.7	5.0	5.3	5.1
SDPS	1.0	1.6	2.6	3.1	3.5	3.7	4.0
NCSM	1.2	1.2	1.7	2.1	2.3	2.2	2.0
Cones [10, 11]							
GC	3.1	3.2	4.1	5.0	5.9	6.1	7.0
BP	3.6	3.6	3.8	4.8	5.7	6.3	6.8
SDPS	3.0	3.1	3.5	4.0	4.5	5.2	6.0
NCSM	3.5	3.5	3.7	4.1	4.5	4.9	5.2
Teddy [10, 11]							
GC	3.0	3.2	3.9	4.5	4.8	5.3	5.5
BP	3.9	3.9	4.8	5.6	6.0	6.5	7.1
SDPS	3.2	4.2	4.9	5.0	6.7	7.2	7.5
NCSM	3.1	3.0	3.2	3.5	4.5	4.7	6.0

These possible matches define candidate volumes of potentially matching points. Only in the final phase, do we fit appropriate surfaces.

(iii) This framework avoids the NP-hard nature of ‘best’ stereo matching for multiple surface scenes and is robust to contrast and offset distortions and high levels of noise. Some known best-matching algorithms are less accurate than NCSM when one image was distorted.

References

- [1] Boykov, Yu., and Kolmogorov, V., An experimental comparison of min-cut / max-flow algorithms for energy minimization in vision, *IEEE Trans. Pat Anal Mach Intell.*, Vol. 26:9, pp.1124–1137, 2004.
- [2] Boykov, Yu., Veksler, O., and Zabih, R., Fast approximate energy minimization via graph cuts, *IEEE Trans. Pat Anal Mach Intell.*, Vol. 23:11, pp.1222–1239, 2001.
- [3] Felzenszwalb, P. F., and Huttenlocher, D. P., Efficient belief propagation for early vision, *Int Jnl Computer Vision*, Vol. 70, No. 1, October 2006.
- [4] Friedman, D., and Drineas, P., Energy minimization via graph cuts: settling what is possible, in *Proc. IEEE CS Conf. Comp Vision and Pat Rec, San Diego, CA, June 20–26, 2005*. IEEE CS Press, Vol.2, pp. 939–946, 2005.
- [5] Gimel’farb, G., Probabilistic regularisation and symmetry in binocular dynamic programming stereo, *Pattern Recognition Letters*, Vol. 23:4, pp.431–442, 2002
- [6] Gimel’farb, G., Liu, J., Morris, J., and Delmas, P., Concurrent stereo under photometric image distortions, in: *Proc. 18th Int. IAPR Conf. Pattern Recognition, Hong Kong, 20–24 August 2006*. IEEE CS Press, 2006.
- [7] Hasler D., Sbaiz L., Süsstrunk S., and Vetterli M., Outlier modeling in image matching, *IEEE Trans. Pat Anal Mach Intell.*, Vol. 25:3, pp.301–315, 2003.
- [8] Liu, J., Delmas, P., Gimel’farb, G., and Morris, J., Stereo reconstruction using an image noise model, in: *Proc. Digital Image Computing: Techniques and Applications (DICTA 2005), Cairns, Australia, Dec. 2005*. IEEE CS Press, pp.476–483, 2005.
- [9] Morris, J., Gimel’farb, G., Liu, J., and Delmas, P., Concurrent stereo matching: An image noise-driven model, in: *Proc. 5th Int. Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition, St. Augustine, Florida, USA, Nov. 6–9, 2005. (Lecture Notes in Computer Science, vol. 3757)*. Springer: Berlin, pp.46–59, 2005.
- [10] Scharstein, D., and Szeliski, R., A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, *Int. J. Computer Vision*, Vol. 47, pp.7–42, 2002.
- [11] Scharstein, D., and Szeliski, R., *Stereo Vision Research Page* [on-line] <http://cat.middlebury.edu/stereo/data.html>, 2006.
- [12] Sun, J., Zheng, N. N., and Shum, H. Y., Stereo matching using belief propagation. *IEEE Trans. Pat Anal Mach Intell.*, Vol. 25:7, pp.787–800, 2003.
- [13] Felzenszwalb, P., and Huttenlocher, D., *Efficient Belief Propagation for Early Vision* [on-line] <http://people.cs.uchicago.edu/~pff/bp/>, 2006.
- [14] Kolmogorov, V., Zabih, R., and Gortler, S., *Graph-cut stereo algorithm* [on-line] <http://www.cs.cornell.edu/People/vnk/recon.html>.

Image Intensifier Characterisation

A.D. Payne¹, A.A. Dorrington¹, M.J. Cree¹, and D.A. Carnegie²

¹ Department of Engineering, University of Waikato, Hamilton.

² School of Chemical & Physical Sciences, Victoria University of Wellington, Wellington.

Email: adp2@waikato.ac.nz

Abstract

An image intensifier forms an integral part of a full-field image range finder under development at the University of Waikato. Operating as a high speed shutter with repetition rates up to 100 MHz, a method is described to characterise the response, both temporally and spatially, of the intensifier in order to correct for variations in the field of view and to optimise the operating conditions. A short pulse of visible light is emitted by a laser diode, uniformly illuminating the image intensifier, while a CCD camera captures the output from the intensifier. The phase of the laser pulse is continuously varied using a heterodyne configuration, automatically producing a set of samples covering the modulation cycle. The results show some anomalies in the response of our system and some simple solutions are proposed to correct for these.

Keywords: Image intensifier, iris, optical gating, range imaging, imaging lidar

1 Introduction

A full field image ranging system is being developed by the authors [1, 2] capable of quickly producing high resolution images by simultaneously measuring the distance to each pixel in the field of view. The system utilises an active light source, typically a number of laser diodes, modulated at frequencies up to 100 MHz. Rather than using a collimated beam and mechanically scanning the field of view, which can take considerable time, this system produces a beam with a much wider angle to flood illuminate the entire field of view. The light impinges on objects in the field of view and is reflected back towards a receiver, consisting of a CCD with a high speed shutter. The shutter is modulated at a slightly different frequency to that of the light source, the difference typically a few hertz or less, and the mixing effect produces a low frequency output, refer Figure 1.

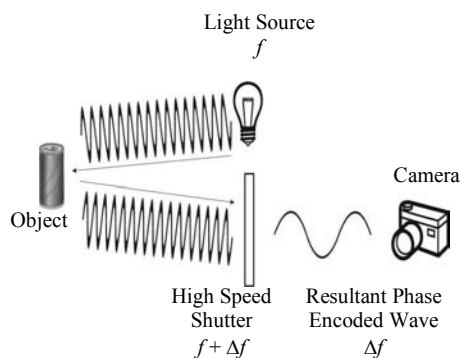


Figure 1: Heterodyning range imager.

The phase of the modulation envelope of the received light is dependent on the distance to the object, and this phase is preserved during the mixing process. A CCD is capable of capturing the low frequency signal (which is below its Nyquist rate) and digitising it. From a minimum of three frames the phase, and therefore range, of each pixel can be calculated. Further detail of the system and some results can be found in a companion paper [1].

The performance of the system is highly dependent on the “high speed shutter” component as the range precision is proportional to the modulation frequency. To achieve frequencies up to 100 MHz an image intensifier is used, with a modulating voltage applied to the photocathode.

1.1 Background

The image intensifier operates by focusing an image on to an input window coated with a suitable photocathode material, in our case S20. When struck by a photon, due to the material’s low work function, an electron is released which is accelerated by an applied electric field. The electron enters a micro channel plate (MCP) where collisions with the MCP walls release secondary electrons, producing a multiplication effect of many orders of magnitude as shown in Figure 2. Upon exiting the MCP, the electrons are again accelerated by an electric field, this time colliding with a phosphor screen. The phosphor emits light, creating an output image which can be viewed by the CCD.

To produce the ‘shutter’ function, the voltage applied to the input photocathode is varied – a negative voltage accelerates electrons into the MCP and hence

produces an output, whereas a positive voltage deflects the electrons away from the MCP and turns the intensifier off. Further detail of the intensifier drive electronics can be found in [3].

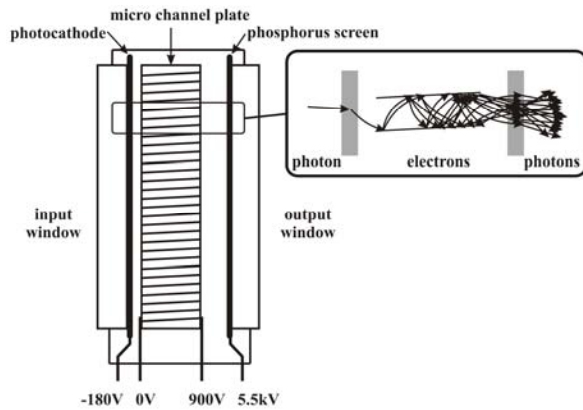


Figure 2: Image intensifier

The modulated voltage is applied to a metal ring around the outer edge of the photocathode and the voltage is conducted through the photocathode material – however the material does have a small resistance. At high modulation frequencies an ‘irising’ effect can occur where the centre of the image intensifier gating is delayed relative to the outer edge [4] due to the resistance of the photocathode and the capacitance between the photocathode and the MCP, forming a low pass filter. In the range imager case this causes a flat object to appear curved, with the centre of the image appearing further away [5].

Electrically measuring the (varying) photocathode voltage is not sufficient to produce information about the image intensifier response as it cannot easily account for this spatial variation. The electrical input to optical output response of the image intensifier is very non-linear, and therefore requires knowledge of this function if a measurement of the electrical signal is to be attempted. It is also worth noting that the capacitive loading by a typical high impedance oscilloscope probe (10-15 pF) would alter the waveform significantly (the photocathode capacitance is approximately 60 pF) and low impedance probes are not suitable for use with the high voltages (50 V peak to peak at frequencies up to 100 MHz).

1.2 Configuration

A simple method of optically measuring the response of the image intensifier has been developed which only requires minimal modification to the original ranging system described above. Instead of operating the laser source with sinusoidal (or 50% duty cycle square) modulation, it is replaced with a pico-second pulsed laser. This pulsed source only illuminates the image intensifier for a very small percentage of each cycle as shown in Figure 3. The output of the image intensifier is integrated over a number of pulses by the

CCD to improve the signal to noise ratio. The produced image is effectively a sample of the image intensifier “shutter” action for that particular point of the cycle.

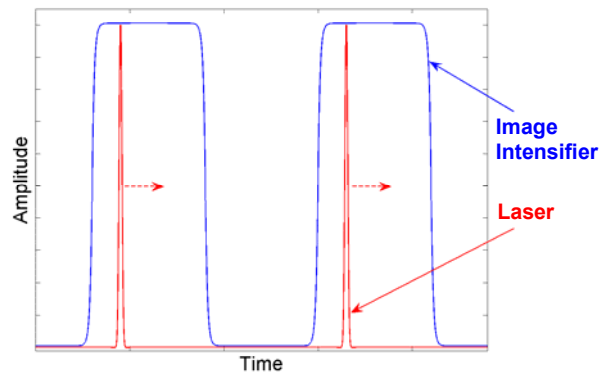


Figure 3: Picosecond illumination of image intensifier

The heterodyne configuration produces a continuously varying phase between the laser pulse and the image intensifier electrical signals, which as indicated by the dash arrow in Figure 3 causes the laser pulse to constantly move through the intensifier shutter waveform taking samples over the entire period. The signal generator also provides a synchronised frame trigger signal to the CCD so that each captured frame occurs at a known phase offset, and a predetermined number of samples can be taken over the waveform period. A similar configuration can be found in [4, 6], although discrete phase steps are manually set between each CCD capture making the process more cumbersome.

2 Laser pulser

Laser pulser systems are readily available from a number of manufacturers, such as the PDL 800-B manufactured by PicoQuant GmbH (Berlin Germany), however the requirements of this configuration make it relatively simple to construct a basic pulser circuit capable of low frequency (less than 100 MHz) repetition rates. Only the pulse width is significant for this experiment as the MCP provides a large gain and therefore high laser peak power is not necessary.

Gain switching a laser diode produces a short optical pulse, down to tens of picoseconds, from a longer electrical pulse [7]. Carriers (electrons) are injected into the active region of the laser, bringing the number of carriers above the lasing threshold. Once above the threshold a large number of photons are produced by stimulated emission within the laser, which in turn reduces the number of available carriers back below the lasing threshold. If the current injected into the laser is turned off at this point, a short optical pulse is generated.

The circuit shown in Figure 4 converts the sine wave input into a short optical pulse. A comparator

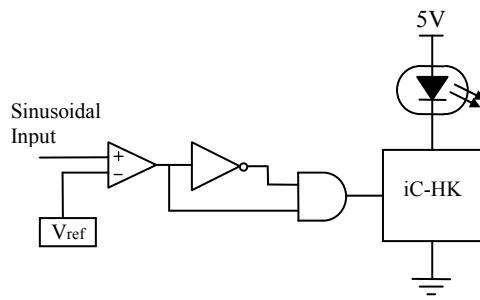


Figure 4: Laser pulser circuit

generates a CMOS level square wave which passes to two inputs of an AND gate, although one input is inverted. Every time the input comparator toggles from low to high, the propagation delay of the inverter ensures both inputs to the AND gate are high for approximately 3 ns, and hence a pulse is produced at the output of the AND gate. An iC-HK laser switch (iC-Haus, Bodenheim, Germany) is used to provide the output drive to the laser diode, and allows the peak current level to be adjusted to optimise the generated output pulse.

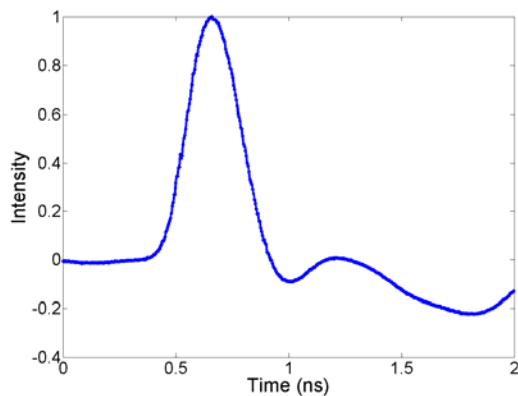


Figure 5: Generated laser pulse, FWHM 266 ps

A recorded pulse is shown in Figure 5 using a Hitachi HL6501MG laser diode. The FWHM pulse width of 266 ps shown is limited by the ~ 2 GHz bandwidth of the photodiode used (Thorlabs SV2-FC), and therefore the laser pulse may be shorter than that shown. For the purpose of this experiment the pulse width here is considered satisfactory as it is significantly shorter than the period of the image intensifier gating.

3 Experimental Configuration and Results

The pulsed laser beam is expanded, and to minimise geometric variation, the image intensifier is placed approximately 1.5 m from the light source so that the light pulse simultaneously illuminates the entire face of the image intensifier. Ground glass is placed in front of the intensifier to remove interference patterns generated by the laser and to ensure the illumination

intensity is uniform. Under normal range finder operation a focusing lens is used in front of the image intensifier, but this is removed for this experiment.

A direct digital synthesiser [8] provides the modulation signal to the image intensifier driver and the laser pulser at a selected frequency with a 0.1 Hz difference, hence it takes 10 seconds for the phase between the laser pulse and the image intensifier gating to cycle through 360° . The output of the intensifier is imaged onto a CCD digital video camera (Dalsa 1M60), which is configured to operate at 100 fps. Therefore 1000 points are captured over the image intensifier period.

3.1 Temporal Response

To measure the ‘shutter’ action of the image intensifier, a small number of pixels in the centre of the recorded image are averaged (to increase the SNR) and are plotted against the frame number as shown in Figures 6 and 7.

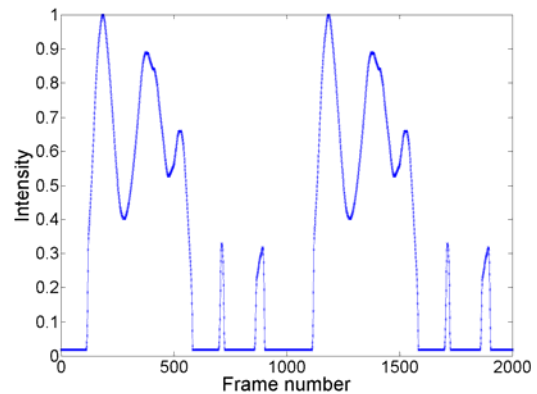


Figure 6: Image intensifier response at 10 MHz

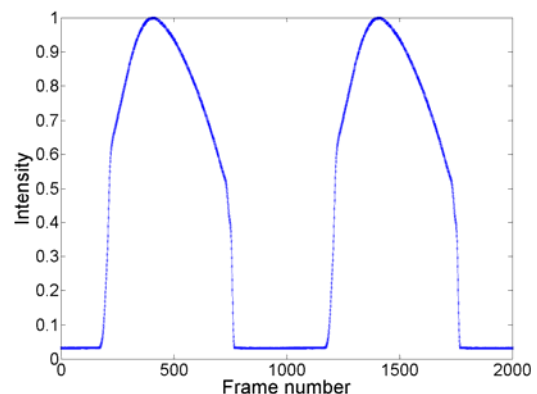


Figure 7: Image intensifier response at 65 MHz

Figure 6 shows that the response is far removed from the desired square wave modulation. Significant ringing occurs due to the capacitance of the photocathode combined with inductance from the interconnecting wires from the electronic driver. During the ‘on’ state the intensity varies up to 60%. During the ‘off’ state the electrical ringing peak is larger than the MCP input voltage, causing the

photocathode to turn on for short pulses. The non-zero output when the intensifier should be in the off state is due to CCD dark current. The resonant frequency is seen to be approximately 50 MHz. The response in Figure 7 approaches the ideal waveform; however the rise and fall times are significantly different. The 10% to 90% rise time is measured to be 2 ns, while the fall time is 3.6 ns.

3.2 DC Response

To understand the optical gain occurring within the image intensifier as a function of the photocathode voltage, a separate experiment was performed. A uniform DC light source is placed in front of the image intensifier, and a DC voltage applied to the photocathode is varied while capturing the image with the CCD. The intensity recorded is graphed in Figure 8.

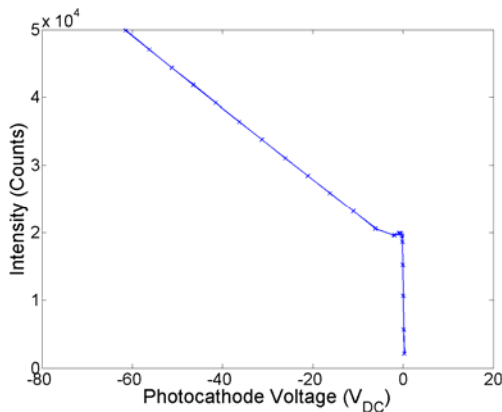


Figure 8: Image intensifier DC response

For a positive photocathode voltage the output drops to zero (slightly above zero in the graph due to CCD dark current). When the voltage becomes slightly negative, a large increase in gain occurs as the electrons emitted from the photocathode are accelerated towards the MCP. From about -2 V onwards the number of electrons reaching the MCP does not significantly change, but each electron receives more kinetic energy from the applied electric field between the photocathode and the MCP, which produces higher gain due to more secondary electrons being produced within the MCP.

The modulation voltage used to drive the image intensifier in section 3.1 was -40 V to +10 V. By looking at the amplitude of the unwanted short pulses in Figure 6, it can be seen from the response in Figure 8 that the ringing after the falling edge is likely to only be slightly negative at its peak, and therefore by altering the bias voltage by a few volts, for example modulating the photocathode voltage from -38 V to +12 V, these extra pulses will be removed.

3.3 Spatial Response

As mentioned in section 1.1, it is possible that the modulation voltage is delayed in the centre of the photocathode compared to the outer edge due to the resistance of the material forming a low pass filter with the capacitance to the MCP. Figure 9 shows this effect, where the intensity of a single captured frame from the rising edge of the waveform is plotted. Instead of a flat surface, a bowl shape can be seen, with the intensity of the centre pixels being less than those at the outer edges.

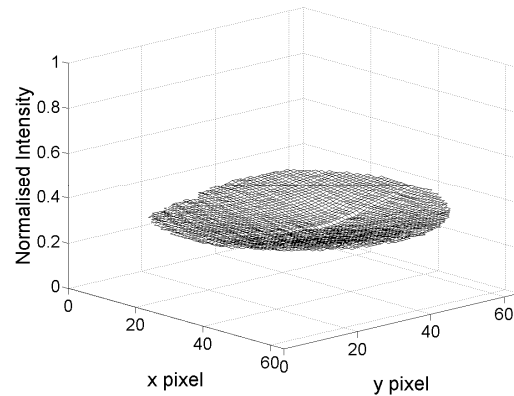


Figure 9: Irising during rising edge at 10 MHz

The iris effect is dependent on the speed of the rising edge transition, and this can be affected by the modulation frequency, therefore the experiment was performed over a range of different frequencies. Figure 10 shows that the iris is much more pronounced at 100 MHz than at 10 MHz as was shown in Figure 9.

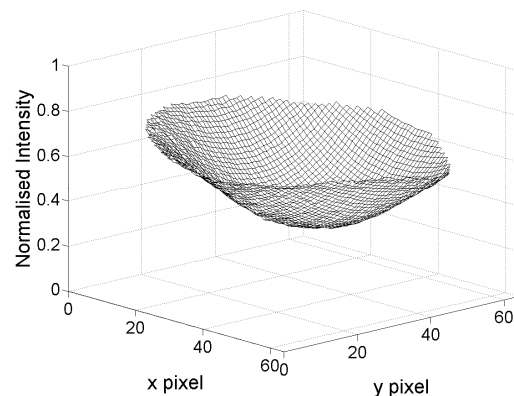


Figure 10: Irising during rising edge at 100 MHz

Figures 9 and 10 show the iris at the most severe point of the waveform (at the sharp rising edge), but it is useful to understand its effect over the entire pulse. Using the assumption that waveform is symmetrical over the round image intensifier, the intensity of a row of pixels through the centre of the image is recorded for each captured frame to represent the entire surface. By plotting this intensity data for various captured frames the iris effect can be seen

as the waveform changes, as shown in Figures 11 and 12. It is worth noting that these graphs are generated by the same data capture as was used to generate Figure 7, however now the plot shows both temporal and spatial information.

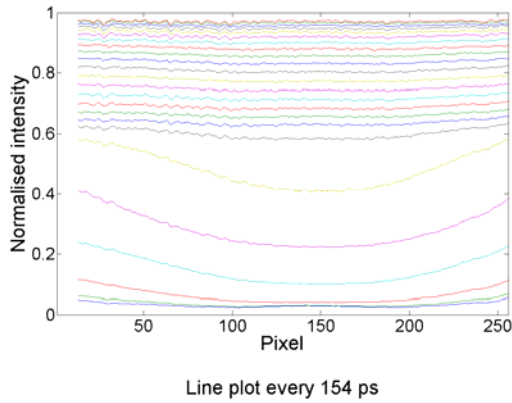


Figure 11: Irising during rising edge at 65 MHz

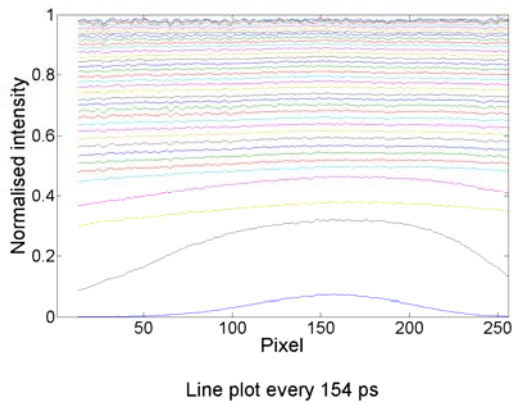


Figure 12: Irising during falling edge at 65 MHz

Figure 11 shows significant irising occurring during a 600 ps interval where a large change in intensity occurs (the lines are separated by large spacing showing a rapid change). The following 2.6 ns illustrate a much slower change in intensity, and as such the level of irising is significantly reduced (although still present). From this figure, the electrical time delay from the edge to the centre of the image is estimated to be 150 ps. The falling edge waveform at the same 65 MHz frequency, refer Figure 12, exhibits different levels of irising to that of the rising edge. The waveform is not symmetrical as can clearly be seen in Figure 7, which leads to this variance due to the different rates of change.

4 Evaluation

By recording the response of the image intensifier and electrical driver, the results indicate that a number of enhancements could be made. The first improvement is to simply adjust the bias voltages so that the electrical ringing after the falling edge cannot turn the image intensifier on as it did in Figure 6. Despite appearing obvious in that figure, it should be noted

that these short pulses are not visible in the captured data when the range imager is running under normal conditions.

The pulse width of the image intensifier optical gating may not necessarily match the duty cycle of the electrical input. Making the negative voltage as large as possible is advantageous as it increases the intensifier gain, refer Figure 8; while a large positive voltage is undesirable as it only increases power dissipation within the system. A sinusoidal input to the photocathode with peaks at +10 V and -40 V is then expected to produce an asymmetrical output which is on 63% of the time (when the photocathode voltage is negative). From the graphs in Figures 6 and 7 a measurement of the duty cycle can be made without knowing the exact characteristics of the electronic pulse (which can often be difficult to measure due to the high voltages and frequencies involved).

Electrical resonance produces significant ringing when frequencies below 50 MHz are used in our system, which distorts the waveform and will therefore introduce an error into the range measurements. In the current configuration, the electrical amplifier output is connected to a second PCB which provides the bias voltages to correctly operate the image intensifier. Redesigning a single PCB to include both the amplifier and bias electronics, as well as reducing the length of the wires to the photocathode, will lower the stray inductance and improve the overall response.

As the magnitude of the irising is dependent on the rate of change of the electrical drive signal, the shape of the waveform becomes important. In a system where the rising and falling edges are not symmetrical the exposure time near the centre of the image may be slightly longer (or shorter) than that near the edge. One possible method to achieve equal rise and fall times is to operate the image intensifier near its resonant frequency, which can be found by observing the oscillation on the rising transition such as that shown in Figure 6. The most significant contribution to the irising occurs when the voltage is very close to zero and the output magnitude is less than 60%. As the output image resolution is also dependent on the photocathode voltage [9], it is desirable to quickly transition through the range near zero volts to produce a higher quality image at the expense of increasing the irising.

In the ranger imager application, the resultant range errors due to irising are independent of the distances measured in the scene. They are dependent on the modulation frequency (as the electrical waveform may not be identical at all frequencies), but are constant for a given frequency and therefore can be calibrated for. The 150 ps delay between the centre and edges of the image (estimated in Section 3.3)

corresponds to a range error of 22 mm, and is consistent with measured errors [5].

5 Conclusion

An image intensifier is used as a high speed optical shutter as part of an image ranging system. An iris effect, where the modulation at the centre of the image is delayed relative to the outer edge, causes the ranger to produce a reconstruction which is curved, with objects at the centre of the image appearing to be at a greater distance than those at the outer edges. Despite the temptation to simply numerically compensate for this effect, it was investigated in depth.

A gain switched laser diode was used to produce picosecond pulses that were temporally scanned across one cycle of the intensifier drive signal. This effectively sampled the image intensifier gating waveform, allowing its optical response to be mapped both spatially and temporally. This temporal scanning was achieved using a heterodyne configuration to continuously alter the phase between the laser pulser and the image intensifier driver. A CCD camera, with a frame trigger synchronised to the other drive signals, was used to capture the image intensifier output.

The experiments revealed that the image intensifier response deviated from the ideal response, most notably with electrical ringing causing a number of problems for low frequency (<50 MHz) operation. This emphasised the fact that the response is dependent on both the image intensifier and the electronic driver as a complete system. Simple variations to the image ranger configuration are proposed to improve its performance, including modifying the driver PCB, adjusting the bias voltages, and selecting the operating frequency at resonance. It is noted that the iris effect cannot be eliminated from the imaging process as it will compromise the image quality, therefore we suggest compensation be added to the image ranger processing software.

6 Acknowledgements

ADP acknowledges the receipt of a TEC Top Achiever Doctoral Scholarship. AAD is funded by a FRST Postdoctoral Fellowship. The authors are grateful to WaikatoLink Ltd. for funding hardware and studentships. The Waikato Imager Ranger is protected by international and New Zealand patents.

7 References

- [1] M. J. Cree, A. A. Dorrington, R. M. Conroy, A. D. Payne, and D. A. Carnegie, "The Waikato Range Imager," *Image and Vision Computing New Zealand (IVCNZ'06)*, (Gt. Barrier Island, New Zealand), November 2006. Accepted.
- [2] D. A. Carnegie, M. J. Cree, and A. A. Dorrington, "A high-resolution full-field range imaging system," *Review of Scientific Instruments*, vol. 76, pp. 083702, 2005.
- [3] A. D. Payne, D. A. Carnegie, A. A. Dorrington, and M. J. Cree, "Full Field Image Ranger Hardware," *Third IEEE International Workshop on Electronic Design, Test and Applications (DELTA'06)*, pp. 263-268, 2006.
- [4] P. Hoss, K. Fleder, and J. Ehrhardt, "Subnanosecond optical gating using coax cable input microchannel plate image intensifier," *Optical Engineering*, vol. 37, pp. 2213-2216, 1998.
- [5] A. A. Dorrington, M. J. Cree, D. A. Carnegie, A. D. Payne, and R. M. Conroy, "Heterodyne range imaging as an alternative to photogrammetry," in *SPIE 6491 – Videometrics IX, San Jose, CA, USA*, January 28 – February 1 2007. Abstract accepted.
- [6] M. Thomas, "Fast optical gating using planar-lead MCPs and linear microstrip impedance transformers," *Proc. SPIE Int. Soc. Opt. Eng.*, vol. 2273, pp. 214-225, 1994.
- [7] K. Y. Lau, "Short-Pulse and High-Frequency Signal Generation in Semiconductor-Lasers," *Journal of Lightwave Technology*, vol. 7, pp. 400-419, 1989.
- [8] A. D. Payne, D. A. Carnegie, A. A. Dorrington, and M. J. Cree, "A Synchronised Direct Digital Synthesiser," *Proc. First International Conference on Sensing Technology (ICST)*, pp. 174-179, 2005.
- [9] L. A. Bosch, "Dynamic uses of image intensifiers," *Proc. SPIE Photoelectronic Detectors, Cameras, and Systems*, vol. 2551, pp. 159-172, 1995.

Noise Models for Symmetric Dynamic Programming Stereo

Zhen Zhou, Georgy Gimel'farb, and John Morris
Department of Computer Science, Tamaki Campus
The University of Auckland, Auckland 1000, New Zealand
{zzho037@ec,g.gimelfarb@,j.morris@}.auckland.ac.nz

Abstract

Symmetric dynamic programming stereo (SDPS) is less accurate on the popular Middlebury stereo data base than the today's best performing stereo matching algorithms. But comparing to them, SDPS is considerably faster and more robust to contrast / offset signal deviations. This paper evaluates to what extent more accurate image noise models and bidirectional processing of a stereo pair can improve the accuracy of the SDPS.

1 Introduction

Dynamic programming stereo (DPS) algorithms have lower accuracy than today's best performing stereo matching techniques such as graph-cut [1] and belief propagation [2] based ones. Nonetheless the DPS is still of sound practical interest due to fast reconstruction of an observed 3D scene and inherent capability to real time processing of still stereopairs and video sequences.

Computational stereo is an ill-posed inverse optical problem with multiple solutions producing just the same stereo pair of images. Human binocular vision relies on visual similarity between images of the same optical surface, and the goal of computational stereo is to select a solution close to human reconstruction, i.e. to approach the visually perceived ground truth. Image similarity depends on an adequate mathematical model of image noise where, in line with [6], the noise is considered as an "umbrella term" embracing basic differences between corresponding signals for every 3-D point in both images of a stereo pair., e.g. pixel-wise random deviations with known or estimated probability distributions due to sensors, spatially constant or variant area-wise contrast / offset deviations, and partial occlusions resulting in image areas with no stereo correspondence. Typically, stereo matching algorithms are derived under quite simple noise models that explicitly account only for pixel-wise random zero-centred deviations and use heuristic thresholds to discriminate between occlusions and large deviations. Only correlation-based matching [5] having been used for many years in digital photogrammetry, the symmetric DPS (SDPS) [3], and recent concurrent stereo matching [6] take explicit account of contrast /

offset deviations typical for real-world stereo images. Experiments show that the SDPS is more robust to such deviations than the best-performing graph-cut and belief-propagation algorithms [4].

One may expect that refinement of noise models on the basis of empirical noise estimates and explicit account for multiple solutions equivalent with respect to stereo matching would reduce the reconstruction errors. One of main sources of errors in the SDPS (as well as in all DP algorithms for stereo matching) is independent reconstruction of each x -oriented epipolar profile from the conjugate scan-lines in the images. Because y -relationships of the signals are not involved, the reconstructed 3D surfaces have typically large "jumps" in y -direction on uniform or repetitive regions where the surface is visually expected to be smooth. The SDPS minimises the total squared distance between the corresponding signals along the scan-lines under the central-symmetric pixel-wise noise, slowly varying limited contrast / offset signal deviations, and explicit partial occlusions of a single opaque surface. Because of many equivalent solutions giving just the same minimum dissimilarity, the independent choice obviously leads to large errors across the scan-lines.

This paper presents initial results of our study of to what extent more accurate noise models and combined bi-directional 3-D reconstruction along and across scan-lines can improve the accuracy of the SDPS. Comparative experiments are conducted with the Middlebury stereo data base [7,8] providing the ground truth (x -disparity maps and occlusion maps) for its stereo pairs.

2 Probability models of signals

SDPS exploits the canonical cyclopean geometry of a stereo pair with image scan-lines parallel to the x -axis. An epipolar profile of an observed 3-D scene is reconstructed from signals of a conjugate pair of the scan-lines by minimising an additive stereo matching score derived from a probability model of these signals [3].

Let $\mathbf{g}_L = (g_L(x_L, y) : x_L \in \mathbf{R}_L)$ and $\mathbf{g}_R = (g_R(x_R, y) : x_R \in \mathbf{R}_R)$ denote 2-D arrays of signals (grey values) for the left and right images of a stereo pair. The images are supported by the finite arithmetic lattices

\mathbf{R}_L and \mathbf{R}_R with integer (x, y) -coordinates, the conjugate scan-lines having the same y -coordinate. 3-D points depicted in the images are represented by their (x, y) -coordinates and x -disparities d in the cyclopean (x, y, d) -lattice such that $x = (x_L + x_R)/2$ and $d = x_L - x_R$, i.e. $x_L = x + d/2$ and $x_R = x - d/2$. For brevity, the y -coordinate is omitted below.

2.1 Conventional noise model

Probability model of the corresponding signals in the SDPS [3] combines independent random signal deviations and interdependent contrast / offset changes along the scan-lines:

$$\begin{aligned} g_L(x+d/2) &= a_L(x)g(x) + b_L(x) + n_L(x) \\ g_R(x-d/2) &= a_R(x)g(x) + b_R(x) + n_R(x) \end{aligned} \quad (1)$$

Here, $g(x)$ is a noiseless cyclopean signal for the point (x, d) , $a_L(x)$, $a_R(x)$ and $b_L(x)$, $b_R(x)$ denote transfer factors and background signals, respectively, describing slow and thus interdependent contrast / offset variations over the images, and n_L and n_R are the independent residual normal random deviations of the signals.

Under natural constraints on the local contrast and offset changes, the model of Eq. (1) results in the sum of squared residual differences between the mutually adjusted corresponding signals as the dissimilarity score for stereo matching [3]. The residual signal difference δ for a binocularly visible point (BVP) in an observed 3-D surface is evaluated after estimating the cyclopean signals g and most likely parameters a , b under constraints preserving, to within a given range, the cyclopean image after its projection onto the stereo pair of images. The constraints apply to relative changes of corresponding signal increments in the left and right projections: if Δg is an increment of the cyclopean signals between the adjacent BVPs along an epipolar profile, then the corresponding increments in the left and right images are $\Delta g_L = \varepsilon \Delta g$ and $\Delta g_R = (2 - \varepsilon) \Delta g$ where $\varepsilon \in [\varepsilon_{\min}, \varepsilon_{\max}]$. The constraints ε_{\min} and ε_{\max} ; $0 < \varepsilon_{\min} \leq 1 \leq \varepsilon_{\max} = 2 - \varepsilon_{\min} < 2$, govern local contrast / offset deviations of one image with respect to the other image as regarding the BVPs. For every partially occluded and thus only monocularly visible point (MVP) in the surface, the matching score includes a positive heuristic weight W_M making the MVPs comparable to the BVPs in the total matching score.

The matching score for the SDPS follows from a Markov chain model of the stereo signals along the conjugate scan-lines representing each cyclopean epipolar profile. Let $\mathbf{d} = ((x_i, d_i, s_i) : i = 1, \dots, N)$ denote such a profile in the symmetric cyclopean coordinates where $s_i \in \{\mathbf{B}, \mathbf{M}_L, \mathbf{M}_R\}$ indicates visibility of the point (x_i, d_i) in the profile (i.e. the BVP or the MVP depicted on the left or right stereo image, respectively). A Markov chain model of the signals is

specified by transition probabilities $p_i(s_i|s_{i-1})$ for the successive points along the profile. For a current BVP $(x_i, d_i, s_i = \mathbf{B})$, the probability depends on a residual signal difference $\delta_{i:\mathbf{B}|s_{i-1}}$ between the corresponding signals for the BVP (x_i, d_i, \mathbf{B}) after their mutual adaptation to account for contrast / offset deviations:

$$p_i(s_i = \mathbf{B}|s_{i-1}) = p_{\mathbf{B}}(\delta_{i:\mathbf{B}|s_{i-1}})$$

The residual difference $\delta_{i:\mathbf{B}|s_{i-1}}$ depends on the preceding visibility state [3], namely, on the adjacent preceding BVP along this particular profile.

2.2 Empirical noise model

Table 1 and Figs. 1 and 2 present empirical marginal probability distributions of signal differences between the corresponding signals for the ground truth correspondence and of signal differences between all possible pixel pairs in the Middlebury stereo pairs "Tsukuba", "Venus", "Cones", and "Teddy". According to these empirical distributions, true stereo correspondences do not necessarily coincide with the closest signal matches. The pairs "Teddy" and "Cones" demonstrate also some offset between the corresponding signals.

Table 1: Empirical distributions (in %) of signal deviations δ for the ground truth correspondences in the "Tsukuba" (Ts), "Venus" (V), "Teddy" (Te), and "Cones" (C) stereo pairs [7, 8] (intervals of δ - a: [-255, -101]; b: [-100, -51]; c: [-50, -21]; d: [-20, -11]; e: [-10, -5]; f: [-4, -2]; g: $\delta = 1$; h: $\delta = 0$; i: $\delta = 1$; j: [2, 4]; k: [5, 10]; l: [11, 20]; m: [21, 50]; n: [51, 100]; o: [101, 255]).

"Tsukuba"								
δ	a	b	c	d	e	f	g	h
Ts	0.5	0.5	1.2	1.7	4.0	17	15	18
V	0.3	1.0	1.6	3.1	6.0	16	16	19
Te	0.3	1.1	2.2	3.6	11	28	13	12
C	0.3	2.4	3.7	6.0	29	30	7.3	5.0
δ	i	j	k	l	m	n	o	
Ts	16	18	4.9	2.0	1.1	0.2	0.0	
V	14	13	5.3	2.6	1.4	0.3	0.0	
Te	9.1	11	4.1	1.9	1.3	0.9	1.1	
C	3.2	4.8	3.6	2.1	1.9	1.2	0.2	

Due to low contrast and offset deviations in the Middlebury stereo images, the adaptation to the transfer factors in the SDPS does not notably change the distribution of the residual signal differences for the BVPs. Figures 1 and 2 show narrow intervals, e.g. $\delta \in [-9, 12]$ for the "Tsukuba" pair, where the empirical probabilities of the true signal differences exceed those for the purely random pairwise differences. The weight $W_M = 50..200$, i.e. $|\delta| = 7..14$, empirically chosen for the SDPS in most

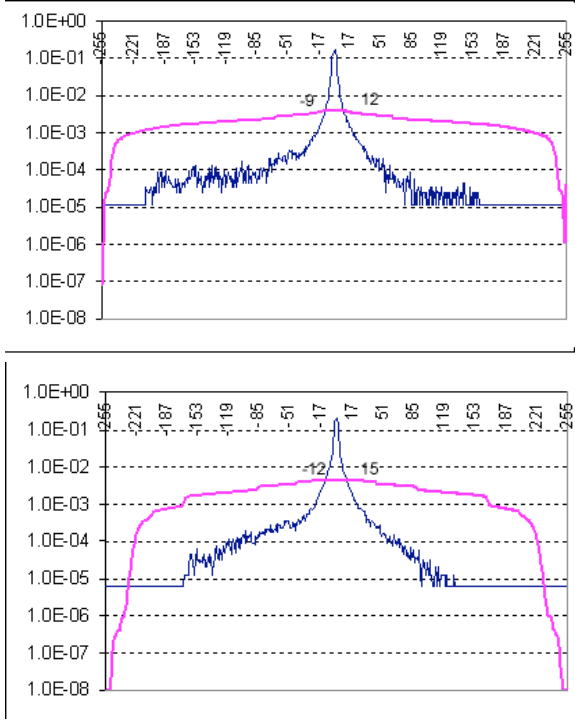


Figure 1: Empirical distributions of differences between the corresponding signals and of all pairwise signal differences for the “Tsububa” (top) and “Venus” (bottom) stereo pairs (the former empirical probabilities exceed the latter ones in the intervals $[-9, 12]$ and $[-12, 15]$, respectively).

of experiments roughly separates the true squared signal differences from the random ones on these four stereo pairs. All the four distributions of the true signal differences are similar but obviously differ from the conventionally assumed normal distribution in Eq. (1). Nonetheless, the normal approximation (i.e. the approximation with a quadratic function $-\alpha(\delta - \delta_0)^2$ for the logarithmic vertical scale in Figs. 1 and 2) may be close to the actual distribution curve at least in the above narrow intervals of the larger empirical probabilities of true signal differences.

The empirical distributions suggest a more natural probability model of stereo images that combines an approximate distribution of residual signal differences δ for the BVPs with a distribution of purely random pairwise differences δ characterising the MVPs. Let $\mathbf{p}_B = (p_B(\delta) : \delta \in [-Q, \dots, 0, 1, \dots, Q])$ and $\mathbf{p}_M = (p_M(\delta) : \delta \in [-Q, \dots, 0, 1, \dots, Q])$ where B and M indicate a BVP and MVP, respectively, and the signals have $Q + 1$ values denote the former and the latter distribution: $\sum_{\delta=-Q}^Q p_B(\delta) = \sum_{\delta=-Q}^Q p_M(\delta) = 1$.

The empirical distributions of random signal differences in Figs. 1 and 2 are almost flat for most of the differences. Hence, the probability of a current MVP (x_i, d_i, \mathbf{M}_L) or (x_i, d_i, \mathbf{M}_R) can relate to the most frequent difference:

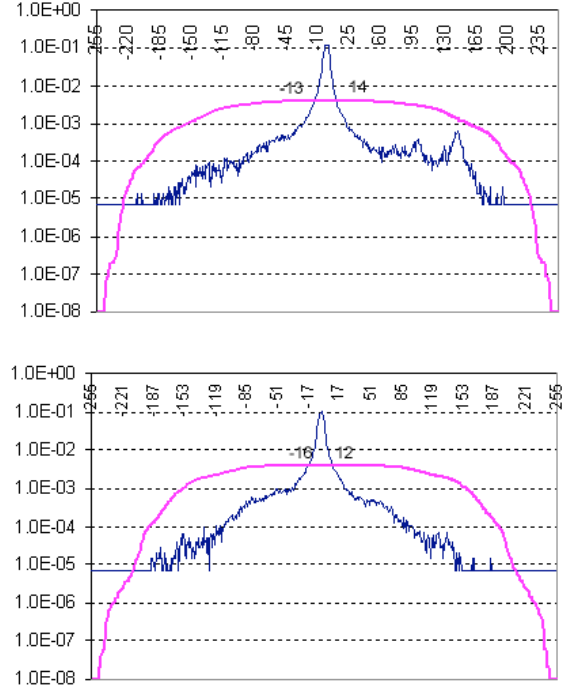


Figure 2: Empirical distributions of differences between the corresponding signals and of all pairwise signal differences for the “Teddy” (top) and “Cones” (bottom) stereo pairs (the former empirical probabilities exceed the latter ones in the intervals $[-13, 14]$ and $[-16, 12]$, respectively).

$$p_i(s_i = \mathbf{M}_L | s_{i-1}) = p_i(s_i = \mathbf{M}_R | s_{i-1}) = p_M^\circ$$

$$\text{where } p_M^\circ = \max_{\delta \in [-Q, Q]} p_M(\delta).$$

The overall probability $P(\mathbf{d}) = p_B(\delta_1) \prod_{i=2}^n p_i(s_i | s_{i-1})$ of signals along a profile \mathbf{d} results in the maximum likelihood stereo matching with the similarity score

$$S(\mathbf{d} | \mathbf{g}_L, \mathbf{g}_R) = \log p_B(\delta_1) + \sum_{i=2}^N \log p_i(s_i | s_{i-1}) \quad (2)$$

to be maximised, or what is the same, the dissimilarity score $D(\mathbf{d} | \mathbf{g}_L, \mathbf{g}_R) = -S(\mathbf{d} | \mathbf{g}_L, \mathbf{g}_R)$ to be minimised for selecting the most likely profile. The DP based optimisation can easily incorporate any empirical probability distributions \mathbf{p}_B and \mathbf{p}_M .

Below we experimentally compare a conventional SDPS with the one where the distributions \mathbf{p}_B and \mathbf{p}_M are estimated for each stereo pair from the known ground truth. The goal is to determine whether a more accurate probability model of the corresponding signals can improve the accuracy of the SDPS comparing to the conventional one with the normal signal noise.

3 Experimental results

Tables 2–4 present the reconstruction results for 3-D scenes in the Middlebury stereo images “Tsukuba”, “Venus”, “Cones”, and “Teddy”. Table 2 compares the SDPS reconstruction with the conventional and empirically estimated noise models. The range for adapting the relative transfer factors is $[0.8, 1.2]$, and the fixed weight for the MVP is $W_M = 50$. Accuracy of the reconstruction along the conjugate scan-lines that accounts for the visibility conditions is practically the same in both cases with a bit better behaviour of the conventional noise model. Seemingly, the only benefit of the use of the empirical signal distribution is in more adequate selection of the weight W_M on the basis of the empirical value of p_M° .

Table 2: Reconstruction accuracy of the SDPS for the conventional (“c”) and estimated (“e”) noise models (the mean, m_a , and maximum, e_m , absolute error and the standard deviation, σ_a , of the absolute error; x – reconstruction along the conjugate scan-lines; y – reconstruction across the conjugate scan-lines in the candidate 3-D volumes estimated by the x -reconstruction).

	m_a	σ_a	e_m	Errors $\leq \theta$; % of points			
				$\theta:0$	1	2	5
“Tsukuba”							
c - x	0.55	1.2	12	71	90	93	99
c - y	0.49	1.1	12	72	92	94	99
e - x	0.56	1.3	13	70	91	93	98
e - y	0.58	1.2	11	70	88	93	99
“Venus”							
c - x	0.64	1.7	19	72	90	93	97
c - y	0.61	1.8	19	75	92	94	97
e - x	0.61	1.7	19	75	92	93	98
e - y	0.59	1.8	19	75	93	94	98
“Cones”							
c - x	1.24	3.2	46	64	84	88	93
c - y	1.12	3.1	46	66	86	89	94
e - x	1.39	3.2	46	61	81	85	92
e - y	1.38	3.1	45	69	81	87	93
“Teddy”							
c - x	1.12	2.7	38	59	85	89	95
c - y	0.94	2.5	39	62	87	91	96
e - x	1.16	2.7	36	58	85	88	94
e - y	1.22	2.6	39	53	82	88	94

Combined stereo matching along and across the scan-lines appears to have much better promise. The conventional SDPS produces not only an output disparity map, but also the residual signal differences δ for all BVPs in the (x, y, d) search space. Just as in the concurrent stereo matching [4, 6], the “optimal” differences for the 3-D points $(x, y, d_{x,y}^*)$ included to the output disparity map \mathbf{d}^* specify the *candidate volumes* of points such that give stereo matches with the same or smaller abso-

lute residual difference $\delta_{x,y,d} \leq \delta_{x,y,d_{x,y}^*}$. Then the SDPS reconstruction across the scan-lines (i.e. in y -direction) is performed only to within the candidate volumes.

Table 3: Reconstruction accuracy (m_a σ_a) of the SDPS for the conventional noise model for the parameters $[\epsilon_{\min}; \epsilon_{\max}]$ and w_M° .

w_M°	Adaptation range $[\epsilon_{\min}; \epsilon_{\max}]$						
	[1.0; 1.0]		[0.9; 1.1]		[0.8; 1.2]		
“Tsukuba”							
50	c-x	0.55	1.2	0.53	1.1	0.55	1.2
	c-y	0.46	1.1	0.48	1.1	0.49	1.1
100	c-x	0.56	1.3	0.55	1.3	0.60	1.4
	c-y	0.47	1.1	0.50	1.1	0.51	1.1
200	c-x	0.65	1.4	0.70	1.5	0.75	1.5
	c-y	0.54	1.2	0.61	1.3	0.65	1.3
“Venus”							
50	c-x	0.77	1.8	0.65	1.7	0.64	1.7
	c-y	0.65	1.8	0.59	1.7	0.61	1.8
100	c-x	0.74	1.7	0.69	1.7	0.69	1.7
	c-y	0.68	1.8	0.67	1.8	0.67	1.8
200	c-x	0.77	1.7	0.75	1.7	0.75	1.7
	c-y	0.75	1.8	0.74	1.8	0.73	1.8
“Cones”							
50	c-x	1.52	3.7	1.28	3.3	1.24	3.2
	c-y	1.26	3.3	1.16	3.1	1.12	3.1
100	c-x	1.58	3.6	1.35	3.3	1.42	3.4
	c-y	1.31	3.3	1.23	3.2	1.27	3.2
200	c-x	1.61	3.5	1.56	3.4	1.63	3.5
	c-y	1.40	3.2	1.37	3.2	1.42	3.3
“Teddy”							
50	c-x	1.25	2.8	1.10	2.6	1.12	2.7
	c-y	0.97	2.4	0.92	2.4	0.94	2.5
100	c-x	1.18	2.6	1.20	2.8	1.30	2.9
	c-y	0.97	2.4	0.95	2.4	1.00	2.5
200	c-x	1.35	2.8	1.43	3.0	1.65	3.3
	c-y	1.08	2.5	1.19	2.7	1.32	2.9

Table 4: Accuracy of the 3-D points that coincide in the surfaces reconstructed by the conventional SDPS along and across the conjugate scan-lines ($p, \%$ – percentage of the coinciding points).

$p, \%$	m_a	s_a	e_m	Errors $\leq \theta$; % of points			
				$\theta:0$	1	2	5
“Tsukuba”							
77	0.27	0.8	12	83	96	97	99.7
“Venus”							
79	0.27	0.9	11	84	96	98	99.5
“Cones”							
74.3	0.67	2.5	46	77	93	94	97
“Teddy”							
69	0.61	2.0	34	72	93	95	98

The y -directed SDPS takes account of only the BVPs. Its transitions and the matching score depend on the corresponding signals: the more uniform are the signals, the smoother should be the surface. Generally, the surface changes have no restrictions in the y -direction, and the surface smoothness is governed by the matching score. Tables 2 and 3 show that the sequential bidirectional matching yields roughly a 10%-improvement of the overall accuracy. The accuracy is slightly better for the adequately chosen reconstruction parameters $[\epsilon_{\min}, \epsilon_{\max}]$ and W_M , but it is still unclear how to evaluate the former range from a given stereo pair. Simultaneously, as shown in Table 4, about 70% of the reconstructed 3-D points coincide for both reconstruction directions, and the accuracy in these areas is (30–50)% higher than the overall accuracy.

To detail these results, Figs. 3 and 4 present the Middlebury stereo pair “Cones” [8] with the ground-truth disparity map for the left image, results of the x -directed and subsequent y -directed SDPS reconstruction of the cyclopean disparity map, and the map of coinciding and non-coinciding disparities for both the reconstruction steps.

This scene contains multiple disjoint surfaces, and some of them are not approximated with a single opaque surface assumed in all the DPS algorithms (e.g. the tops of cones and sticks in the mug that violate the ordering constraint). These errors appear in both disparity maps in Fig. 4,a-b because the candidate volumes for the y -directed stage are specified at the x -directed stage. The SDPS, as well as any stereo algorithm reconstructing a single surface under the ordering constraint, cannot escape these errors.

At the same time positions where the disparities differ in the both reconstructed maps (Fig. 4,c) relate mostly either to object boundaries (places of rapid changes of disparities) or to boundaries between areas of constant disparity. This suggests a promising direction for the further work - to merge both the disparity maps by propagating the coinciding disparities (if necessary, with interpolation) to the “non-coinciding” areas.

4 Conclusions

The above experimental results show that there are no significant improvements in accuracy if the conventional simple signal model in Eq. (1) is refined by using empirical estimates of the true probability distributions of signal differences. At most, the latter distributions allow us to more naturally select weights for partially occluded points in the stereo matching score, yielding only marginal changes in the reconstruction accuracy.

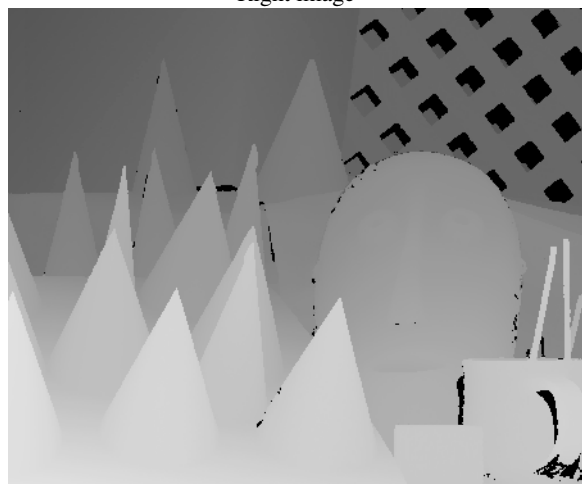
The bidirectional SDPS reconstruction where the second, y -directed stage is constrained to the candidate volumes found at the first, x -directed one offers a more



Left image

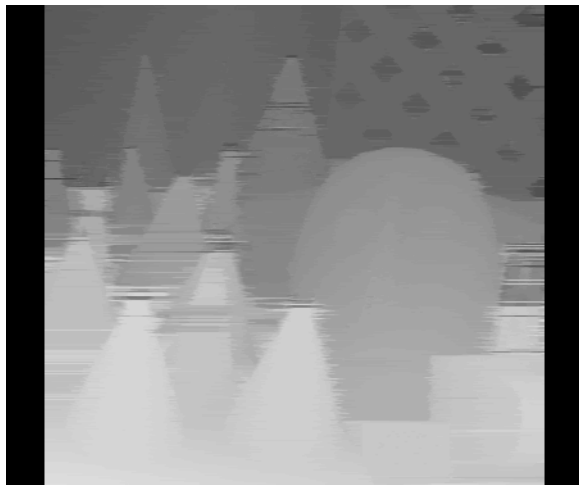


Right image



Ground-truth disparity map
(black – points with unknown disparities)

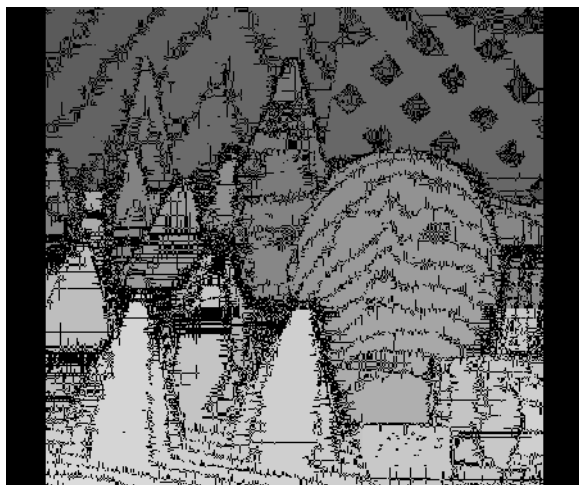
Figure 3: Stereo pair “Cones” [8] (tops of cones and sticks violate the ordering constraint assumed in the DPS).



a: Disparity map along the scan-lines (x)



b: Disparity map across the scan-lines (y)



c: Non-coinciding (black) disparities in both the maps

Figure 4: SDPS reconstruction for "Cones".

considerable promise. Although the reconstruction in principle produces a single surface and thus cannot avoid errors for objects that violate the ordering constraint, the whole process outputs about 70–75% of the whole scene with the twice better accuracy than the unidirectional SDPS. Because the remaining areas mostly separate the “more accurate” ones, there is a good reason to expect that the former can be filtered out by expanding the latter.

References

- [1] Boykov, Yu., and Kolmogorov, V., An experimental comparison of min-cut / max-flow algorithms for energy minimization in vision, *IEEE Trans. Pattern Analysis Machine Intell.*, Vol. 26:9, pp.1124–1137, 2004.
- [2] Fetzenszward, P. F., and Huttenlocher, D. P., Efficient belief propagation for early vision, *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition (CVPR 2004), Washington, DC, USA, 27 June - 2 July 2004*. IEEE CS Press: Los Alamitos, Vol. 1, pp.261–268, 2004
- [3] Gimel'farb, G., Probabilistic regularisation and symmetry in binocular dynamic programming stereo, *Pattern Recognition Letters*, Vol. 23:4, pp.431–442, 2002
- [4] Gimel'farb, G., Liu, J., Morris, J., and Delmas, P., Concurrent stereo under photometric image distortions, *Proc. 18th IAPR Int. Conf. Pattern Recognition (ICPR 2006), Hong Kong, China, 20–24 Aug. 2006*. IEEE CS Press: Los Alamitos, Vol. 1, pp.111–114, 2006.
- [5] Helava, U. V., Object-space least-squares correlation, *Photogrammetric Engineering and Remote Sensing*, Vol. 54, pp.711–714, 1988.
- [6] Morris, J., Gimel'farb, G., Liu, J., and Delmas, P., Concurrent stereo matching: An image noise-driven model, in: *Proc. 5th Int. Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR 2005), St. Augustine, Florida, USA, Nov. 6–9, 2005. (Lecture Notes in Computer Science, vol. 3757)*. Springer: Berlin, pp.46–59, 2005.
- [7] Scharstein, D., and Szeliski, R., A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, *Int. J. Computer Vision*, Vol. 47, pp.7–42, 2002.
- [8] Scharstein, D., and Szeliski, R., *Stereo Vision Research Page* <http://cat.middlebury.edu/stereo/data.html> [on-line] 2006.

Tracking iris surface deformation using Elastic Graph Matching

Sammy S.S. Phang¹, Wageeh Boles¹ and Michael J. Collins²

¹School of Engineering Systems, Queensland University of Technology, Brisbane, Australia

²School of Optometry, Queensland University of Technology, Brisbane, Australia

Email: s.phang, w.boles, m.collins@qut.edu.au

Abstract

We propose a method to track the iris surface deformation in image sequences captured by a special infrared illuminated high-speed camera using elastic graph matching. A circular grid elastic graph (*iris graph*) to track the iris radial and circular movement of pupillary activity caused by varying lighting conditions is introduced. We compare the phase similarity with the magnitude similarity for tracking iris features and we also relate the determination of the weight of topography preservation in the similarity cost function to the amount of movement of the object being tracked. The algorithm is tested with a series of synthetic iris images and a series of real iris images. We show that the deformation of the iris surface area during the constriction of the pupil operates mainly in the middle and the peripheral parts of the iris and that this deformation is non linear.

Keywords: Iris movement, iris recognition, elastic graph matching, Gabor filters, tracking

1 Introduction

The role of the iris has become increasingly important compared to many other biometrics in many human recognition systems [1]. This is because of the exceptionally unique characteristics of the irises of every individual. However, one of the problems this technology still needs to address is its sensitivity to variations in the pupil size [2]. Since our pupil size fluctuates all the time as it responds to the ambient brightness conditions, chances of capturing images of the same person with a different pupil size are high [3]. This would affect the performance of iris recognition systems. For example, in a verification scenario, if the iris image captured on the spot registers a pupil size very different from the pupil size of the original iris image captured during the enrolment, the verification may fail.

There are several ways to deal with the problem. One way is to ensure that the iris image captured during the recognition stage has the same as the pupil size as the one captured during the enrolment stage. However, this is not a feasible solution because of the constant fluctuation of pupil size with the existence of light. Furthermore, the size of the pupil is also controlled by other factors. For example, a drowsy person or a person who has been affected by certain drugs will have a different pupil size [3]. Another way to address the problem is to study and model the physiology behaviour of iris surface deformation for various irises. Such a model can be integrated into iris recognition systems to improve their performance.

In this paper, we introduce a method for iris surface deformation tracking using the Elastic *Graph Matching* (EGM) algorithm. This algorithm was

initially proposed for translation invariant object recognition [4]. This algorithm has also been successfully applied to face and gesture recognition [4, 5]. The robustness to varying face position and facial expressions (e.g. smile, cry, and laugh) of EGM algorithm has inspired us to use it to track a deformable object like the iris surface. The algorithm then uses the convolution coefficients of an image with a family of Gabor wavelets of different frequencies and orientations to compare the similarity between two objects. These convolution coefficients are referred as Gabor wavelets' responses. EGM has also been extended to *Morphological Elastic Graph Matching* for face tracking purposes [6], where instead of using the Gabor wavelets' responses, it uses responses from various morphological operations.

In brief, *EGM* algorithm is a basic process to compare graphs with images and to generate new graphs. A single labelled graph is matched onto an image. This labelled graph has sets of convolution coefficients extracted from the image by a family of wavelets where each set is centred on one image point. The sets are referred to as jets and are arranged in a particular spatial order. The image jets initially have the same relative spatial arrangement as the graph jets, and each image jet corresponds to one graph jet. The similarity of the graph with the image is simply the average jet similarity between image and graph jets. The graph is allowed to translate, scale and distort to some extent, resulting in a different selection of image jets to increase the similarity. The distortion and scaling is limited by a penalty term in the matching cost function [5]. Our experiments show that this

penalty term plays an important role for the elastic iris graph.

The paper is organized as follows. In the next section, we provide the details of our iris surface tracking method. In section 3, experimental results are presented and discussed. The conclusions are given in section 4.

2 Method for Iris Surface Movement Tracking

The iris surface tracking process consists of four steps, as follows:

- (i) Pre-processing the iris image to determine the iris parameters.
- (ii) Constructing the elastic iris graph and locating the graph on the iris in the image.
- (iii) Performing image Gabor transformation.
- (iv) Tracking the iris surface deformation between two consecutive images in a sequence using elastic graph matching.

This section describes these four steps.

2.1 Iris Image Pre-processing

The iris images to be processed are captured from a video sequence. This consists of a series of iris images of increasing or decreasing pupil size. In some cases, unwanted eye blinking may be captured in a video sequence. We eliminate those images by exploiting the fact that the total grey level value of an image with the blink is much higher than the one without the blink. By assuming the first few images are good quality images (i.e. consist of sufficient iris surface area for tracking), we calculate an average sum of the grey level values, A , of these images from the video sequence, as in equation (1). The sum of the grey level values of current image being processed is calculated using equation (2). We exclude the image if its average grey level value is larger than a certain threshold, t , given by equation (3).

$$A = \frac{1}{N} \sum_{i=1}^N \left(\sum_{x=1}^{nx} \sum_{y=1}^{ny} I_i(x, y) \right) \quad (1)$$

Where ny and nx are the rows and columns in each image. N is the number of images and we used $N=5$ in our experiments.

$$B = \sum_{x=1}^{nx} \sum_{y=1}^{ny} I_{current}(x, y) \quad (2)$$

where $I_{current}$ is the current image being processed.

$$t = \frac{abs(B - A)}{B} \quad (3)$$

For each iris image, we determine the centre and the boundary of the limbus and the pupil. We use our previous work in [7] for this purpose. In summary, we first find the initial centre of the eye by exploiting the circular symmetry property of the pupil. Then, we

identify the limbus sectors (which is the area of an iris between the upper and lower eyelid that contains the transition from the limbus to the sclera). Next, we transform the image into a polar coordinate representation and determine the limbus and pupil edges by zero crossing. We use the detected candidate points of the limbus and pupil to estimate the parameters of the pupil and limbus models. We model the limbus with a circle and model the pupil boundary with an ellipse.

2.2 Elastic Iris Graph

An elastic graph is a set of *nodes* connected by *edges*. The edges are used to code the topography (i.e. where the features of interest are located) and are labeled with distances. Each node is labeled with a jet. Such a local description of, for example, a specific iris feature can be used to search for the same or a similar feature in the subsequent image (we give more about this in section 2.3). Thus, the geometry of an object is encoded by the edges while the grey value distribution is patchwise encoded by the nodes.

An elastic graph can take various geometry structures. A rectangular grid graph or a face graph is generally used in face recognition. The nodes of a face graph are located at points of interest such as face contour, eyes, nose and lips. Since the iris has a circular shape and its movements are in the radial and circular directions, to track these two movements effectively, we introduce an *iris graph*, which is a non-regular grid graph as shown in Figure 1.

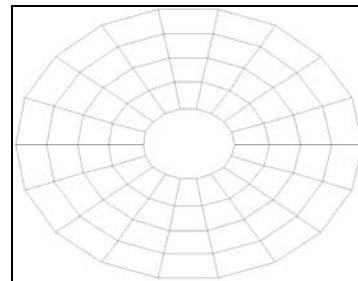


Figure 1. Example of an 18x5 nodes iris graph.

The graph nodes are the intersection points between radial lines (separated by a fixed angle θ°), and piecewise linear concentric circles (separated by equal intervals, r , in pixels). The origin of the system is placed at the centre of the iris. Each meridian has the same number of nodes. We determine the number of nodes by dividing the smallest radial iris length with a fixed radial spacing (in pixel). Then, we equally space the node radially for each meridian from the limbus to the pupil edge. Figure 1 shows an example of such a graph consisting of 18x5 nodes, with $\theta = 20^\circ$ and $r = 20$ pixels.

2.3 Gabor Transformation of Images

As mentioned in the previous section, each node of the elastic graph is a jet and EGM uses jets between

two images to determine their similarity. We extract the jet from the Gabor wavelet transformation of an image. Since Gabor transformation is computationally expensive, we reduce the processing time by limiting to the area containing the iris. We use the limbus centre to register and align the position of the eye.

A Gabor wavelet is a complex sinusoid multiplied by a two dimensional Gaussian. When a function is convolved with a Gabor wavelet, the frequency information near the centre of the wavelet is captured, and frequency information far away from the centre of the Gaussian is filtered out. In order to describe the frequency information of a local feature in an image accurately, it is necessary to convolve the pixel values of that location of an image with a variety of Gabor wavelets. Gabor wavelets can take a variety of forms. We employ the Gabor formulation of Petkov and Kruizinga [8] and this may be written as:

$$g_{\lambda,\theta,\varphi,\sigma,\gamma}(\vec{\chi}) = \exp\left(\frac{x'' + \gamma^2 y''}{2\sigma^2}\right) \cos\left(2\pi \frac{x''}{\lambda} + \varphi\right) \quad (4)$$

$$x'' = x \cos \theta + y \sin \theta ; y'' = -x \sin \theta + y \cos \theta \quad (5)$$

where $\vec{\chi} = (x, y)$ specify the position of a light impulse in the visual field. The parameters $\lambda, \theta, \varphi, \sigma, \gamma$ of equation (4) are the wavelengths, orientations, frequency offsets, standard deviation of the Gaussian factor and the spatial aspect ratio of the Gabor wavelet respectively. For simplicity, let us denote $g_j(\vec{\chi}) = g_{\lambda,\theta,\varphi,\sigma,\gamma}(\vec{\chi})$, with the subscript, j , refers to the combination of these parameters.

Let $I(\vec{\chi})$ be the grey level distribution of the input image. Convolving a Gabor function, $g_j(\vec{\chi})$, with the image at location $\vec{\chi}$ gives a jet $J_j(\vec{\chi})$ that describes a small patch of grey values around that pixel location:

$$J_j(\vec{\chi}) = \int I(\vec{\chi}) g_j(\vec{\chi}) \quad (6)$$

The integral here produces complex coefficients $J_j(\vec{\chi})$ of a jet that consists of real (a_j^r) and imaginary (a_j^i) parts. This complex coefficient can be represented in polar coordinates having a total magnitude of $a_j(\vec{\chi})$ and phase angle of $\phi_j(\vec{\chi})$ and using equations (7) and (8).

$$a_j = \sqrt{(a_j^r)^2 + (a_j^i)^2} \quad (7)$$

$$\phi_j = \begin{cases} \arctan(a_j^i / a_j^r) & a_j^r > 0 \\ \pi + \arctan(a_j^i / a_j^r) & a_j^r < 0 \\ \pi / 2 & a_j^r = 0 \text{ \& } a_j^i \geq 0 \\ -\pi / 2 & a_j^r = 0 \text{ \& } a_j^i < 0 \end{cases} \quad (8)$$

2.4 Tracking Using Elastic Graph Matching (EGM)

The idea behind of the EGM algorithm is the concept of Dynamic Link Architecture. It exploits the correlations in the fine-scale temporal structure of cellular signals, in order to group neurons dynamically into higher order entities, where these entities represent a very rich structure [4, 5]. The graph matching algorithm tries to find a position for each node of the graph which maximizes the feature similarity and minimizes the topography costs at the same time. The rigidity or flexibility of the graph can be determined by weighting the topography costs in the overall cost function.

The EGM algorithm consists of two phases. The first phase is called a *Global Move*, where we try to approximate the best matching position by not allowing distortion of the graph. This means that each time we move the graph on the image we are moving all nodes uniformly. The second phase is called *Local Move*, where we allow each node to move individually to a new position around its search neighbourhood that has the maximum feature similarity.

Before we go into the mathematical details of the EGM algorithm, let us define the first iris image frame presented to the algorithm to be I and the next frame with slightly deformed iris to be I' . With the assumption that the iris surface deformation is small (e.g. 4ms interval between two frames), EGM algorithm is able to identify the most matching position of each of the nodes of the iris graph in the new image. We use the distorted iris graph from the previous tracking as the initial iris graph for the tracking of subsequent frames. The direction and magnitude of displacement information of each node is used to decide certain parameters (describe below) of EGM tracking algorithm.

In matching a landmark in image I to I' , we use two similarity functions: a similarity based on Gabor wavelet response (S_v) and a similarity based on the geometry topography (S_e) and a cost function C_{total} introduced in Martin *et. al.* works [3]:

$$C_{total} = \kappa C_e + C_v \quad (9)$$

where

$$C_e = \sum_{(ji) \in E} S_e(\Delta_{ij}, \Delta'_{ij}) \quad (10)$$

$$C_v = -\sum_{j \in v} S_v(J_j, J'_j) \quad (11)$$

C_v is the cost relating to the Gabor jet similarity and C_e is the cost relating to the connecting edges at the nodes of the iris graph. The κ in equation (9) controls the topography of the elastic graph. Small κ values allow the graph to distort while large κ values make the elastic graph more rigid. The choice of κ depends

on the amount of movement or deformation. If the movement between frames is large, then a smaller κ value should be used. We discuss the choice of κ in section 3.

During the matching, we try to preserve the topography between the iris graph in I and iris graph in I' . This is imposed by allowing minimum change to the edge distance of the connecting nodes. The connection between nodes $\vec{\chi}_i$ and $\vec{\chi}_j$ in iris graph is labelled as Euclidean distance vector:

$$\Delta_{ij \in E} = \vec{\chi}_i - \vec{\chi}_j \quad (12)$$

where E is the set of edges in the iris graph. The labels of the iris graph in I are compared to those in I' by a quadratic comparison function, S_e :

$$S_e(\Delta_{ij}, \Delta'_{ij}) = (\Delta'_{ij} - \Delta_{ij})^2 \quad (13)$$

The square term ensures that the Euclidean distance is positive value and helps to differentiate the nodes with small Euclidean distance from those nodes with large Euclidean distance from their connecting nodes. The similarity based on Gabor wavelet response, S_v , can be divided into two measures: magnitude similarity (S_a) and phase similarity (S_ϕ). The first measure's results is a similarity measure based on the covariance of the magnitudes:

$$S_a(J, J') = \frac{\sum_j^M a_j a'_j}{\sqrt{\sum_j^M a_j^2 \sum_j^M a'^2_j}} \quad (14)$$

where M is the number of Gabor wavelets, while a_j and a'_j are the magnitude of Gabor wavelet response in polar coordinate (see equation 7) for image I and I' respectively.

This method is tolerant of small displacement. It measures the energy of the frequency responses and is unaffected if the frequencies are out of phase. Hence, we use this similarity for *global move* to position the iris graph in image I' more accurately. In *global move*, we let the topography of the graph to be unchanged. Thus, C_e is zero in this case. Since the phase information is excluded, the measure can be easily confused and may respond to an incorrect spatial feature. Hence, we use the second similarity measure (S_ϕ) for *local move*, to improve the localisation of the nodes of the iris graph in image I' .

A Gabor wavelet responds strongly to edges if the direction is perpendicular to its wave vector, but when hitting an edge, the real and the imaginary parts oscillate with the characteristic frequency instead of providing a smooth peak. Since phase varies rapidly with displacement, jets taken from an image a few pixels apart from each other have very different coefficients, although they represent almost the same local feature. This allows us to discriminate between

patterns with similar magnitudes. Indeed, the phase similarity measure also based on the magnitude response, but these values are weighted by the similarity of phase angles. Thus high scores are achieved only when both the magnitude and phase angle are similar. This measure effectively computes a similarity between -1.0 and 1.0.

$$S_\phi(J, J') = \frac{\sum_j^M a_j a'_j \cos(\phi_j - \phi'_j)}{\sqrt{\sum_j^M a_j^2 \sum_j^M a'^2_j}} \quad (15)$$

During the *local move* graph distortion, with the assumption that the displacement is not too big, we define a search area of $(\pm\Delta x, \pm\Delta y)$ around each node for iris graph in I' . For simplicity of explanation, let us consider matching a node at $\vec{\chi}$ in I to I' only. We compute the S_ϕ and S_e around the defined neighbourhood of $\vec{\chi}$ in I' , and the cost C_{total} for each pixel in the defined search neighbourhood area. We identify the pixel position that constitutes a local minimum of C_{total} as the most matching pixel location to node $\vec{\chi}$ in I . Each node in the elastic graph is visited sequentially and in random order until all the vertices in the graph have found their new position. The label (i.e. the edge's Euclidean distance) vector is updated dynamically.

The distorted iris graph is then used as the initial graph for I' and I'^{+1} . The direction and amount of displacement of each node in previous tracking is used to estimate the displacement of the node in next frame. The amount of displacement is also used to determine the size of the search neighbourhood and the parameter κ of a node in the next frame tracking.

3 Experimental Results

We tested the proposed method on a sequence of synthetic iris images with dilating pupil size. The circular band with darker grey colour is the pupillary region and the circular band with lighter grey colour is the ciliary iris region. The size of the synthetic iris image is 256x256. The iris graph consists of 4x5 nodes, which has a radial spacing of 20 pixels apart starting from the limbus edge and 20° spacing starting from 150° to 210° as measured from the positive x-axis with the origin at the centre of the iris, as shown in Figure 2a. In first test, we dilated the pupil by five pixels for each frame and moved the features in a radial direction linearly. All the features have been moved about three to ten pixels in their radial direction, with features closer to the pupil edge moved more than features closer to the limbus edge. The radius of the neighbourhood search size should be at least twice the size of the expected object's displacement. In this case, we also tried using a denser iris graph of 4x10 nodes (as shown in Figure 3a) with a radial spacing of 10 pixels and circular

spacing of 20° starting from 150° to 210° to determine the sensitivity of the algorithm to the grid size.

In the second test, we introduced nonlinear circular movement of the middle two synthetic iris features by additional $4^\circ, 6^\circ, 12^\circ$ apart. We used an iris graph of 4×10 nodes for this test. In third test, we evaluated the sensitivity of the algorithm by tracking the features in every other frame and compared this result with the result of frame-by-frame tracking. Finally, we tested our algorithm with a series of real iris images.

The parameters of the Gabor wavelets used in this algorithm are:- $\lambda \in \{4, 4\sqrt{2}, 8, 8\sqrt{2}, 16\}$, $\theta = \mu\pi/8$ where $\mu = 0, \dots, 7$, $\gamma = 1$ and $\varphi = \{-\pi/4, \pi/4\}$ where $\varphi = -\pi/4$ is thought to be the real part of the wavelet and $\varphi = \pi/4$ is thought to be the imaginary part of the wavelet. This created two wavelet masks that are mirror image of each other. We used a circular wavelet support ($\gamma = 1$) to ensure the wavelet responses of a node has an equal effect of surrounding movement of the node. The size of the wavelength λ depends on the resolution of the image and how much surrounding features we want taken into account. Reducing the λ value helps the local features to stand out during image localisation, while increasing this value help in tracking the movement of homogeneous regions. This yields eight orientations, five frequencies, and two phases for a total of 80 different wavelets (40 complex convolution values).

As mentioned in section 2.4, the value of κ depends on the amount of movement of the node and the value of S_e as given by equation (13). Since S_e is a value between -1 to 1, then κ should be a factor that brings S_e to a value between 0 and 1. This depends on the amount of distortion allowed for a node. For the synthetic images, we used $\kappa = 10^{-3}$ for nodes that have distortion less than 10 pixels and $\kappa = 10^{-5}$ for nodes that have distortion of 10 pixels or more.

Test one tracking results are shown in Figure 2. Most of the nodes of the iris graph are located on the edge of the synthetic iris features. We can notice from the results that most of the nodes are remained at the same edge location of the synthetic iris features after tracking. The tracking result is compared with their known displacement, and we found errors of less than 1.5 pixels for most of the nodes. The result of the second test is shown in Figure 3. The algorithm is able to track non-linear circular object's movement. However, as we can notice, the algorithm fails to track the middle iris feature at 210° of Figure 3d. This is because the algorithm tracks the middle iris feature at 150° first then updates its new location. This new position has increased the edge distance between the node of the feature at 210° and the feature at 150° ,

and this large change has caused the algorithm to fail to track that feature. In order to address this problem, the phase similarity should be given more weight than topography similarity by setting a smaller κ value for that node. We did not find any significant difference for the third test between the results of frame-by-frame tracking and every other frame tracking. The two tracking results are identical for these synthetic images. Therefore, the algorithm is not sensitive to its initial graph for small displacements.

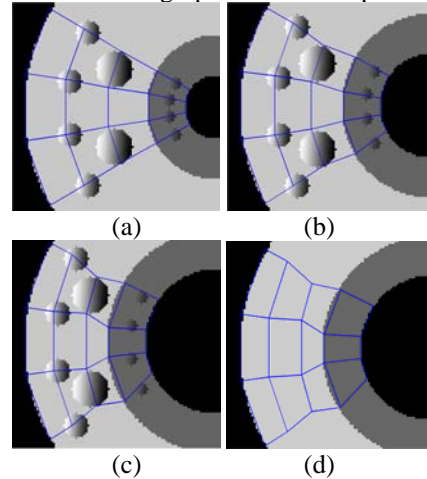


Figure 2. Results of the 1st test. Iris features are moved linearly in the radial direction from (a) to (c). Figure (d) shows the final deformed graph.

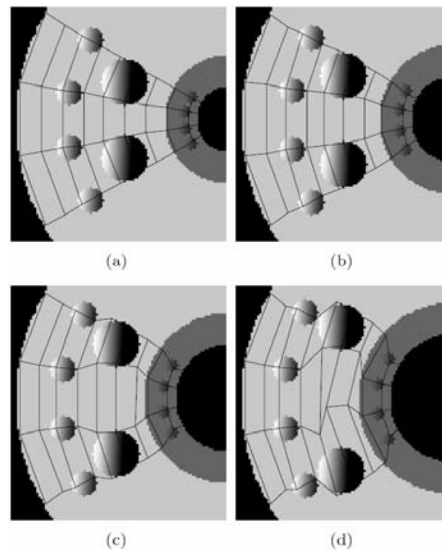


Figure 3. Results of the 2nd test. The two middle iris features are moved non-linearly in circular direction from (a) to (d).

Figure 4 gives the tracking results of a series of real iris images. We only performed the tracking on the lower right quadrant of the iris surface. The full image resolution is 1024×1024 and the image size after pre-processing and cropping is 400×400 . We used $\kappa = 10^{-3}$ for nodes within the ciliary iris region (i.e. radial nodes one to five from the limbus edge) and $\kappa = 10^{-5}$ for nodes within the pupillary region (i.e.

radial nodes six to eight from the limbus edge). We used an iris graph of 8x8 nodes with a radial spacing of 30 pixels apart and 10° spacing starting from 280° to 350°. We can see that the graph's nodes located at the iris features remained attached to the nodes during the tracking. We found that there is no significant cyclo-rotation of iris surface movement. We also found that the iris surface area increases during constriction are mainly from the middle and the peripheral parts of the iris. This gain seems to be linear until the pupil loses its ellipticity, as we can notice the surface gain of peripheral iris area between 280° ~ 300° is greater than the surface gain of peripheral iris area between 310° ~ 350°.

4 Conclusions

In this paper, a method for tracking 2D iris surface movement using Elastic Graph Matching is presented. The algorithm uses an iris graph that allows us to track radial and circular movements of iris features. The initialization of the iris graph is based on the results of the pre-processing stage. We tested the algorithm on a series of synthetic iris images with known movements and the algorithm gives an overall tracking error of less than 1.5 pixels compares to their known displacements. We also tested the algorithm with a series of real iris images. We found that the iris extends its surface area from the middle part of the iris during pupillary constriction. The obtained tracking results show that this method is able to track the iris surface deformation during pupillary activities.

This is an on going research and for future work, a method to determine an appropriate value for κ based on the values of radial edge distance and elliptical edge distance (i.e. the edge distance between two nodes separated by a large angle) to encounter the problem of significant change in edge distance of circular connecting nodes.

5 Acknowledgments

The authors would like to thank Dr. Ted Dunstone for his suggestion of using the EGM approach and many useful discussions and Weiling M. B. for his help with the Gabor filter algorithm.

6 References

- [1] T. Mansfield, G. Kelly, D. Chandler, and J. Kane, "Biometric product testing final report," tech. rep., Centre for Mathematics and Scientific Computing, National Physical Laboratory, Queens Road, Teddington, Middlesex, TW11 0LW, 2001.
- [2] M. Li, Y. H. Wang, and D. X. Zhang, "Efficient iris recognition by characterizing key local variation," *IEEE Trans. Image Processing*, vol. 13, no. 6, pp. 739–750, 2004.
- [3] I. E. Loewenfeld, *The pupil - anatomy, physiology, and clinical applications*, vol. 1, ch 1. The iris. Butterworth-Heinemann, 1999.
- [4] M. Lades, J. C. Vorbrüggen, J. Buhmann, J. Lange, C. Malsburg, R. P. Wurtz, and M. Konen, "Distortion invariant object recognition in the dynamic link architecture," *IEEE Transactions on Computers*, vol. 42, pp. 300–311, 1993.
- [5] L. Wiskott, J. M. Fellous, N. Krüger, and C. Malsburg, "Face recognition by elastic bunch graph matching," in *Intelligent Biometric Techniques in Fingerprint and Face Recognition*, ch. 11, pp. 355–396, CRC Press, 1999.
- [6] G. N. Starmou, N. Nikolaidis, I. Pitas, "Object tracking based on morphological elastic graph matching", *IEEE International Conference on Image Processing, 2005. ICIP 2005. vol 1*, pp: 709-12, 11-14 Sept. 2005
- [7] S. S. S. Phang, D. R. Iskander, and M. J. Collins, "High speed pupillometry," in *Proceedings of the 8th Australian and New Zealand Intelligent Information Systems Conference, CISRO*, pp. 43-47, Sydney, December 2003.
- [8] N. Petkov and P. Kruizinga, "Computational models of visual neurons specialised in the detection of periodic and aperiodic oriented visual stimuli: Bar and grating cells," 1997.

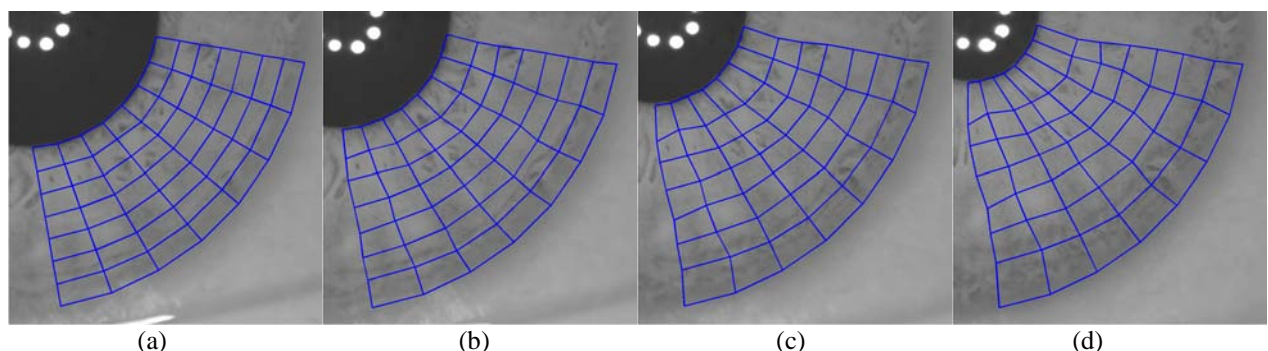


Figure 4. Results of a series of real iris images tracking during pupil constriction. Progression of graph deformations from (a), with large pupil, to (d) small pupil size.

Perceptually Correct Image Space Soft Shadows

R. Rountree¹, R. Rayudu¹ and D. Brebner²

¹ Massey University, Institute of Information Sciences & Technology

² Unlimited Realities, Palmerston North

r.k.rayudu@massey.ac.nz

Abstract

Soft shadows generated from area or line light sources create an added sense of realism to a scene while also providing spatial hints about the scene. In this paper we present a new real-time algorithm that generates soft shadows by blurring in image space while reducing the artefacts currently associated with an image space blur. The penumbra region is estimated and only the estimated region is blurred, thus reducing the time taken for the blur stage. This algorithm creates perceptually correct soft shadows in real time on modern graphics hardware.

Keywords: Soft shadows, shadow algorithm, real-time shadows, shadow mapping

1 Introduction

Shadows play a very important role when producing images using computer graphics applications [1]. Recent advances in computer graphics (CG) and CG hardware have made real-time 3D graphics a reality [2]. However, computing high-quality, realistic shadowing for dynamic scenes in real time is a difficult task and has generated considerable research interest [2].

This paper describes a new real-time algorithm that creates perceptually realistic soft shadows. Our algorithm utilises depth maps and blur maps and creates soft shadow maps from blur maps. The result is a perceptually correct blur as shown in Figure 1.

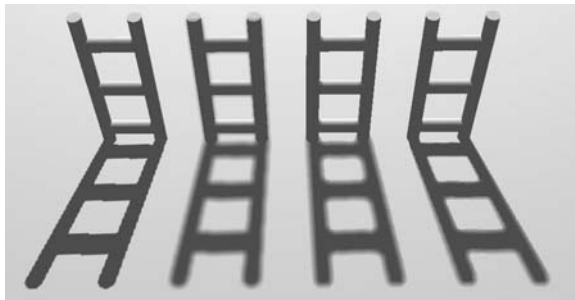


Figure 1: Left to right: hard shadows, uniform blur, restricted blur, perceptually correct blur.

Shadow mapping and shadow volumes create hard-edged shadows which are only made by infinitely small point light sources. Line and area light sources create soft-edge shadows which depend on the size of the light and the distance between the light source, blocker and receiver.

Soft shadow algorithms have been developed, in which the soft edges are created by sampling in world space. An image space blur was proposed by [8] in

which hard shadows were cast and the resulting image was blurred in image space and then sent on to lighting calculations. This created uniform soft shadows in which the penumbra region did not vary. There are also two predominant artefacts associated with blurring in image space.

This paper is organised as follows: section 2 discusses related research and section 3 describes the problem in detail. In section 4, we introduce our algorithm for creating soft shadows. Section 5 details our implementation and the results are discussed in section 6. We present our results in several example scenes rendered by our algorithm and they are illustrated in the Appendix that follows our conclusion.

2 Related Research

Shadow mapping was introduced by Williams in 1978 [3]. In his process the scene was first rendered from the lights point of view and the depth of the scene stored as a depth map. The resulting depth map was then projected onto the scene and used to determine whether the result was in light or shadow. However, this technique created only hard-edge shadows. Since this time, there have been many proposals to create soft-edge shadows by expanding on this method.

Flavien Brebion [8] proposed in 2003 a technique to blur hard shadows in an image using shadow volumes [4]. This technique was prone to certain artefacts which will be described in detail later. Anirudh Shastry [9] later demonstrated that the same results could be achieved using shadow mapping. These two techniques take into account the shadows cast and blur the result in image space before being used in the final lighting calculations.

Eric Shan and Frédo Durand proposed the Smoothies algorithm [10] in which additional geometry was

created in image space to identify penumbra regions and create soft-edged shadow. Wyman and Hansen's Penumbra Map algorithm [11] was similar; however the additional geometry was created in world space not image space.

Péter Tamás Kovács and György Antal proposed a method [19] of image space blurring in which the most prominent artefact, the halo, was removed from the scene. This was done through testing surrounding depth values in image space to find edges so that the blurring would not sample incorrect regions. However, the method was still affected by the other artefacts associated with the image space blur.

Our algorithm is based on shadow mapping and progresses on from the work of Flavien Brebion [8] in which the blur is done in image space without requiring any additional geometry to be added to the scene.

3 Our Research Problem

The most prominent visual artefact created by the uniform image space blur is the halo or bleeding effect. This occurs when an area of shadow or light incorrectly affects a neighbouring region of opposite value. The radius of the blur directly affects how noticeable this artefact is. When only a small blur is used the artefact is mostly hidden; however this means that large penumbra regions cannot be created. When a large blur radius is used to simulate a large area light source the effect becomes too large to be ignored. This artefact is most prominent in areas where there is a high concentration of alterations between areas of shadow and light.

The uniform image space blur produces only shadows with a fixed penumbra region. This does not show the effects when the distance between the light, blocker and receiver changes. This is an important aspect to include when creating soft shadows, as varying penumbra regions provide useful spatial information hints.

Another artefact of the uniform image space blur is that when viewing a plane on a sharp angle the blur radius is effectively increased. Due to the image space sampling translating into a larger sample area in world space, shadows viewed at sharp angles become faded out.

The last artefact is only noticeable when the viewer is moving around a scene. As the uniform image space blur does not factor in the depth of the scene during the blurring stage, objects further away from the viewer will appear to have a larger penumbra region. As the viewer moves closer to a penumbra region it will decrease in size. This is mostly noticeable in very large scenes spanning great distances.

In this research, these problems were solved by the projection of extra depth maps, to create an estimated penumbra region along with an umbra region. By blurring only within the penumbra region, the halo artefact was removed. The extra projected shadows also created a perceptually correct variation in penumbra. The umbra region, which is not blurred, prevented the entire shadow becoming washed out when viewed at low viewing angles. Finally a depth value was used to modify the blur radius to prevent distant shadows becoming too blurred.

The following figures demonstrate some of the problems discussed here and a solved output from our algorithm. Figure 2 shows the effect of blur radius from a lower angle. The image on the right shows the result from our algorithm.

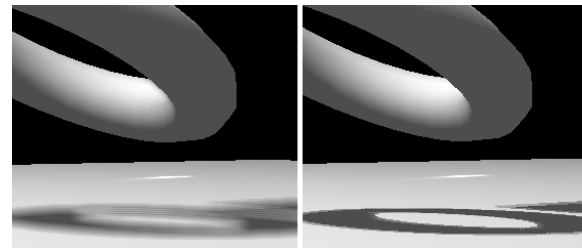


Figure 2: Problem related to viewing angle

Figure 3 shows the halo effect as discussed in the initial part of this section. The image on the right shows the result of our solution.

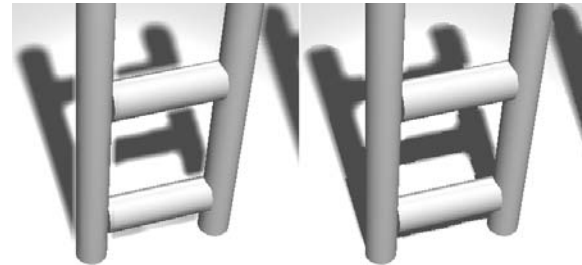


Figure 3: The halo effect problem

4 The Algorithm

Our algorithm has the following passes:

Pass 1: As with traditional shadow mapping, the first step is to render the depth map from the lights viewpoint. Our algorithm creates four depth maps, one from each corner of the area light source, all in parallel.

Pass 2: The four depth maps are then projected onto the scene at the same time from the corners of the light source. It is important to note here that at this stage no lighting calculations are performed. The result is a greyscale image where each projection casts a light value of 0.25 and a dark value of 0. This is encoded into a blur map utilising 3 colour channels.

The first channel stores the penumbra region estimate and is given a value of 1 when the greyscale map has values 0.25-0.75. The second channel is used to store the data which is to be blurred. The fully lit value is assigned when the greyscale image is 0-0.5 and a fully occluded value is given when the greyscale image is 0.75-1. The final channel is used to store a depth value for the camera. The depth value needs to be normalised to be stored correctly in the resulting image.

Pass 3: The next stage is to blur the blur map. The blur is only run inside the estimated penumbra region. The radius of the blur is then divided by the value stored in the third channel of the blur map, the depth function value. The result of this blur is saved as the soft shadow map, which shows the regions of shadow in image space.

Pass 4: The resulting image is then used in the final lighting pass where the scene is rendered for the final time. The screen coordinates of the final rendering are used to find the value stored in the soft shadow map. The value found is multiplied against the diffuse and specular terms while the ambient and emissive terms remain unchanged.

The restricted image space blur is very similar to the perceptually correct image space blur. The difference occurs in the first stage where only one depth map is rendered from the centre of the light source. The single depth map is then projected from the corners of the light source. The rest of the restricted blur algorithm is exactly the same as the perceptually correct blur.

Figure 4 shows the pictorial representation of our algorithm. A more detailed pictorial depiction after each pass of our algorithm is shown in figure 5.

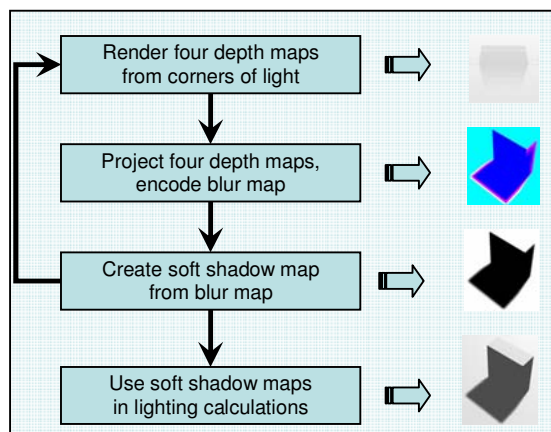


Figure 4: The four passes of our algorithm.



Figure 5: Results of the four passes.

5 Implementation

The implementation of our algorithm was done in OpenGL with CG shaders. The machine used was an Athlon 64 3200+ with 1 GB RAM. The graphics card was a NVIDIA 7900GT 256MB, which supports dynamic branching in shaders. The algorithm benefits most when run using dynamic branching. However, it is still able to perform in real time on lower end hardware.

The implementation used only traditional shadow mapping so it was not omnidirectional; however, this could easily be extended through the use of cube mapping or dual parabolic shadow mapping [12].

Our results show that the halo artefact is now reduced to within the penumbra region. The estimated umbra region is always fully shadowed. The bleeding of darkness into lit areas has also been completely removed. With the perceptually correct blur the penumbra region is seen to change based on the distance between the light source, blocker and receiver. The restricted blur does not do this due to the projection of the same depth map from different points. The result is visually less believable but there is a significant performance gain in using only one depth map, not four as in the perceptually correct blur.

The size of the light is not the only factor required by this algorithm. A uniform base value is required for the radius of the blur. This value is relative to the size of the light source and can be altered in real time to find the optimal value for variable sized light and resolution of the image space blur texture.

Our implementation was built to work on a wide range of graphics hardware. The OpenGL command `glCopyTexSubImage2D` was used to store the result of the different render passes and shadow maps. Rendering directly to texture through the use of pixel buffers would be a more optimal solution and increase the speed of our implementation.

6 Results

Three different factors were taken into account during benchmarking: resolution, polygon count and number of blur samples. Resolutions of 512x512 and 1024x1024 were used to see the effects of the blur code, which are dependent on the size of the texture to blur. Scenes with low (656) and high (211298) polygon counts were used to find the effect of the increased number of render passes required for the perceptually correct blur.

Table 1: Experimental results.

Resolution	Polygon Count	Blur Samples	Hard Shadows (frames/sec)	Uniform Blur	Restricted Blur	Perceptually Correct Blur
1024	High	Low	126	110	96	58
1024	Low	Low	324	235	187	154
512	High	Low	161	154	145	78
512	Low	Low	>999	844	676	552
1024	High	Medium		84	80	53
1024	Low	Medium		116	156	135
512	High	Medium		136	132	74
512	Low	Medium		424	553	467
1024	High	High		51	56	42
1024	Low	High		48	109	99
512	High	High		106	109	67
512	Low	High		179	400	369

We experimented with different sample sizes for the blurring pass and found that a 7x7 box blur provided a good balance of speed vs. quality. The number of samples used in the blur has a significant impact on the speed of the algorithm.

Table 1 shows the achieved results of our tests. The blur samples in the table were 3x3 (small), 7x7 (medium) and 11x11 (high) box blurs. The frame rates for hard shadows were the same for the rest of the tests and hence are not listed. Our results demonstrate that when there are a high number of blur samples, the restricted and perceptually correct blurs show significant performance gains over the uniform blur through dynamic branching. The shadows created by each method are depicted in figures 6 and 7. These figures, along with figure 1, demonstrate that our algorithm out-performs the other methods.

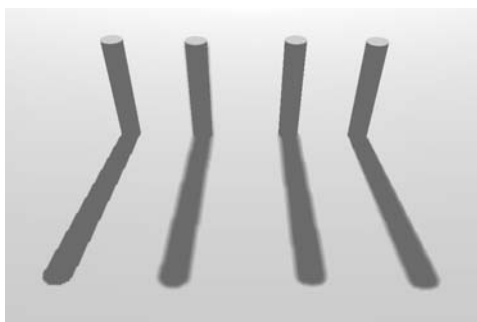
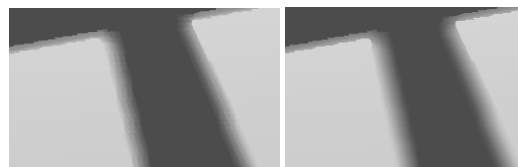
**Figure 6:** Left to right: hard shadows, uniform blur, restricted blur, perceptually correct blur**Figure 7:** Left to right: hard shadows, uniform blur, restricted blur, perceptually correct blur

Figure 8 shows close-up shots of soft shadows to demonstrate the effects of different blur samples. There is a considerable difference between 3x3 and 11x11 samples.

**Figure 8:** Soft shadows: 3x3 blur sample (left) and 11x11 blur sample (right).

The restricted blur was found to create a large number of artefacts when the size of the light was increased. This is due to the use of only one depth map for the projection. The restricted blur is therefore recommended for small light sources only.

The perceptually correct method can handle larger lights without difficulty; however, when the size of the light is very large or the blocker is very small, both techniques fail. When the umbra region is very small or nonexistent there are no dark values to blur, resulting in the shadow disappearing.

7 Conclusion

We have described a new algorithm to create soft shadows in real time using existing graphics hardware. Our algorithm creates perceptually correct soft shadows as an image space technique. We have identified the most prominent problems in creation and depiction of soft shadows and either removed or substantially reduced these effects in our algorithm. We demonstrated that our algorithm performs better and faster than other common algorithms and achieves real-time performance.

Subsequently, using the programmable features of graphics hardware, we will be improving our algorithm towards more realistic and real-time results.

Creating an omnidirectional solution that better utilises modern graphics hardware is planned for the future.

8 Acknowledgements

This project is jointly funded by Unlimited Realities and the New Zealand Government's TIF grant.

Thanks to Robert Grapes from Unlimited Realities for his knowledge of OpenGL. Thanks to Russell Brebner of Unlimited Realities for his help and support.

Tutorials from nehe.gamedev.net have been very helpful.

9 References

- [1] T. Akenine-Moller and E. Haines, *Real-time Rendering*, A K Peters Ltd., 2nd ed., 2002.
- [2] J. Hasenfratz, M. Lapeirre, N. Holzschuch and F. Sillion, *A Survey of Real-time Soft Shadows Algorithms*, STA Report, Eurographics 2003.
- [3] L. Williams, "Casting curved shadows on curved surfaces", *In Proc. SIGGRAPH*, vol. 12, 270–274, 1978.
- [4] F. Crow, "Shadow algorithms for computer graphics", *In Computer Graphics (Proc. SIGGRAPH)*, vol. 11, 242–248, 1977.
- [5] F. C. Crow, "Summed-area tables for texture mapping", *H. Christiansen, Ed.*, vol. 18, 207–212, Minneapolis, 1984
- [6] L. Williams, "Pyramidal parametrics", *In Computer Graphics (SIGGRAPH '83 Proceedings)*, 1–11, 1983.
- [7] Wimmer, M., Scherzer, D., and Purgathofer, W., "Light space perspective shadow maps", *In Proceedings of the 2nd EG Symposium on Rendering*, Eurographics Association, Springer Computer Science, Eurographics, 2004.
- [8] Flavien Brebion: Soft Shadows. *ShaderX2 : Shader Programming Tips and Tricks with DirectX 9.0*, Wolfgang F. Engel (editor), Wordware Publishing, Inc., 2003
- [9] Anirudh Shastry, "Soft-Edged Shadows", <http://www.gamedev.net/reference/articles/article2193.asp>
- [10] Chan, E. and Durand, F. 2003. Rendering fake soft shadows with smoothies. In *Proceedings of the 14th Eurographics Workshop on Rendering* (Leuven, Belgium, June 25 - 27, 2003). ACM International Conference Proceeding Series, vol. 44. Eurographics Association, Aire-la-Ville, Switzerland, 208-218.
- [11] Chris Wyman and Charles Hansen, "Penumbra maps: Approximate soft shadows in real-time". *In Rendering Techniques 2003 (14th Eurographics Symposium on Rendering)*. ACM Press, 2003.
- [12] S.Brabec, T.Annen and H.Seidel, "Shadow Mapping for Hemispherical and Omnidirectional Light Sources", *In Proc. of Computer Graphics International.*, 2002.
- [13] Agrawala, M., Ramamoorthi, R., Heirich, A., and Moll, L. Efficient image-based methods for rendering soft shadows. *In Proceedings of ACM SIGGRAPH 2000*, ACM Press / ACM SIGGRAPH / Addison Wesley Longman, Computer Graphics Proceedings, Annual Conference Series, 375– 384. ISBN 1-58113-208-5, 2000.
- [14] J.C. Russ, *The Image Processing Handbook*, Florida: CRC Press, 2nd ed., 1995.
- [15] Brabec, S., and Seidel, H.-P. Single Sample Soft Shadows Using Depth Maps. *In Proc. Graphics Interface*, 219–228, 2002.
- [16] McCool, M. D. *Shadow volume reconstruction from depth maps*. ACM Transactions on Graphics 19, 1 (January), 1–26. ISSN 0730-0301, 2000.
- [17] Woo, A., Poulin, P., and Fournier, A. A survey of shadow algorithms. *IEEE Computer Graphics & Applications* 10, 6 (November), 13–32, 1990.
- [18] Heidmann, T. *Real shadows real time*. IRIS Universe, 18, 28–31, 1991.
- [19] Péter Tamás Kovács and György Antal, *Soft-Edged Stencil Shadow in CAD Applications*, Third Hungarian Conference on Computer Graphics and Geometry, Budapest, 2005.

Hardware implementation of the Maximum Subarray Algorithm for Centroid Estimation

S.J. Weddell, B.N. Langford

University of Canterbury, Dept. Electrical & Electronic Engineering.

Email: steve.weddell@canterbury.ac.nz

Abstract

This paper discusses a hardware implementation of the maximum sum subarray algorithm and an extension for centroid estimation suitable for Shack-Hartmann wavefront sensors. Maximum efficiency is achieved by utilising a systolic architecture for the estimation of centroids. The maximum subarray algorithm is offered as an alternative to conventional centroid estimation using a VLSI hardware implementation. Our approach offers an efficient method for the calculation of centroids over multiple regions, directly in hardware. Other possible applications of a hardware implementation of the maximum sum algorithm include medical imaging and machine vision. To meet the high computational demands of such applications a highly efficient hardware implementation, optimised for a parallel algorithm, is described.

Keywords: FPGA, image processing, maximum subarray centroid estimation, wavefront sensor

1 Introduction

To meet the high computational demands required for the compensation of atmospheric turbulence in real-time, hardware implementation of image processing algorithms is required. For example, the estimation of several thousand centroids over an image frame, typically updated at a frequency of 1 kHz, is required for specialised image sensors used in adaptive optics [1].

Hardware solutions employing mixed-signal technology have been developed for the estimation of centroids in real-time [2]. Our approach is to develop a centroid estimator using the maximum subarray algorithm in hardware and aims to improve computational efficiency on current methods.

Our hardware implementation of the maximum subarray algorithm has allowed us to attempt problems that require efficient centroid estimation. An example is the embedded hardware required to process images obtained from wavefront sensors used in adaptive optics.

Section 2 provides a background discussion on topics covered in this paper. Section 3 outlines a hardware implementation of the maximum subarray algorithm. This is followed by a summary of our simulations in Section 4. Our results are presented in Section 5. A suggested application is given as future work in Section 6 and this is followed by our conclusion in Section 7.

2 Background

In this section an overview on astronomical wavefront sensors is provided. This is followed in Section 2.2 by a brief explanation of a commonly used algorithm for centroid estimation. An overview of the maximum sum algorithm is given in Section 2.3

2.1 Astronomical Wavefront Sensors

Adaptive optics (AO) is a technology used to counter the adverse effects of a distorting medium, such as a turbulent atmosphere in astronomy [1].

A typical AO system will use one or more wavefront sensors, in conjunction with a closed-loop control system, to enable actuators that drive deformable mirrors. Such mirrors are used to alter the optical path of a telescope imaging an object. The conjugate of the distorting wavefront can be used to correct optical aberrations within an image, in real-time.

A critical component of any AO system is the wavefront sensor. Due to their simplicity of operation, the Shack-Hartmann wavefront sensor is typically used to estimate the centroids of spatially segmented reference objects, through the provision of multiple, sub-apertures [3]. Wavefront slopes measured as centroid data are taken from each sub-aperture. This resulting centroid data set is used by a reconstruction algorithm to estimate

3.1 Field Programmable Gate Arrays

Field Programmable Gate Arrays (FPGAs) comprise a fabric of combinational and sequential logic, Configurable Logic Blocks (CLB), and block RAM modules. Input Output Blocks (IOBs) provide an interface to external devices. FPGAs, as distinct from Complex Programmable Logic Devices (CPLDs), are generally more suited to the implementation of arithmetic functions due to the provision of carry chain logic and additional support for sequential logic configurations.

Our initial development used the XC2VP2 device from the Virtex-II Pro series of Xilinx FPGAs to implement the maximum subarray algorithm. This device provided a total of 1,408 slices (352 CLBs), 216Kb of block RAM and 204 IO pads.

3.2 Centroid Estimations

Current hardware implementations of centroid algorithms for wavefront estimation [2], are comprised of arithmetic modules such as adders, multipliers and dividers, arranged in a structure similar to Figure 2.

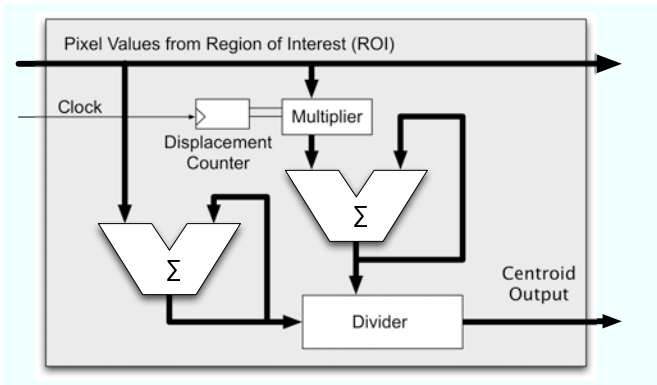


Figure 2: Centroid Hardware Implementation [2]

For FPGA implementation, Equations 1 and 2 in Section 2.2 are implemented in hardware using the arithmetic modules supplied by the HDL component libraries. Efficient methods exist for implementing modules that require multiplication and division, however, accumulation is an iterative process. For example, Figure 2 shows two accumulator modules, accumulating one pixel every clock cycle.

As outlined in Section 2.3, the maximum subarray algorithm efficiently finds the rectangular subarray of the maximum sum within a given 2-D array. In addition, the location of the subarray, using two co-ordinates to define the region of the subarray, is provided.

Given an imaging application where each cell shown in Figure 1 represents a pixel value, v ,

the algorithm is well suited for the estimation of centroids for two reasons.

Firstly, given a point source object at infinity, δ , projected through a medium comprising several perturbing layers of atmosphere, and diffracted by telescope optics, the resulting distorted image ψ , when formed on an image plane and captured by an high frame-rate camera, defines an ROI, Λ . Multiple sub-regions, ζ are created within Λ when a Shack-Hartmann wavefront sensor is used. To acquire wavefront data, a continuous series of centroid estimates are required from each sub-region, ζ . The resulting image is similar to that shown in Figure 4, where each sub-region requires concurrent, centroid estimation.

Within each sub-region, if the midpoint is found between the co-ordinates produced by the maximum subarray algorithm, shown as L in Figure 3, then x and y centroids within ζ can be easily determined. Thus, the x -centroid, \bar{x} and y -centroid \bar{y} are given as,

$$\bar{x} = \frac{r_2 - r_1}{2}, \quad (3)$$

$$\bar{y} = \frac{c_2 - c_1}{2}, \quad (4)$$

where r_1, c_1 is the first co-ordinate, and r_2, c_2 is the second co-ordinate of the maximum sum subarray shown as an output at cell, $P_{k \times j}$ in Figure 2.

Secondly, the time complexity of the maximum sum algorithm, over an $N \times M$ array, is $O(n)$ time, based on $O(n^2)$ processors. In terms of actual performance, this is equivalent to, $T_{(N+M-1)}$ clock cycles. This performance metric, however, is only possible once the systolic array is fully loaded. A detailed discussion on concurrent processing and loading of systolic arrays for k -maximum sum values is given by Bae [7].

3.3 Maximum subarray Algorithm

Our work comprised of defining a cell structure using the Very High Speed Integrated Circuit (VHSIC), Hardware Description Language (VHDL), and generating a lattice structure that propagates the maximum sum v , and the location of the maximum subarray, $(r_1, c_1), (r_2, c_2)$, from one of the four apex cells located around the perimeter of the structure, to an adjacent apex cell on the opposite side of the structure. For example, this is shown in Figure 3 as $P_1(1, 1)$ to $P_{k \times j}(N, M)$.

The cell entity outlined in Section 2.3 and shown in Figure 1, was implemented using separate control and datapath modules and was modelled using

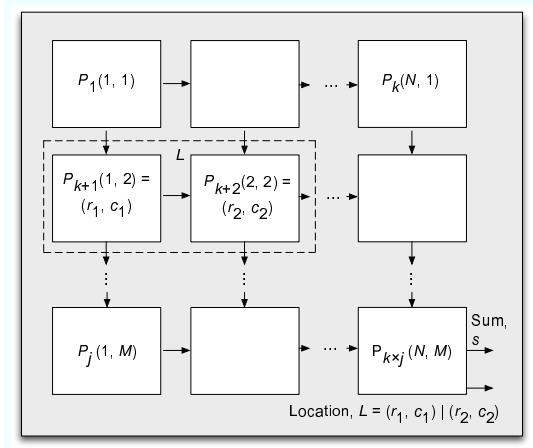


Figure 3: The Cell Array

both behavioural and structural VHDL. To ensure optimum performance, a highly concurrent implementation of the control unit was required.

Initially, a 2×2 cell matrix was configured and tested to demonstrate the efficiency of a hardware implementation of the maximum sum algorithm. This initial cell configuration was increased incrementally by 2^n cell entities to a maximum size, where $n=8$. For each increment, the utilisation of FPGA resources was assessed. Details of these results are provided in Section 5. Test-bench waveforms were written to verify correct operation and these are provided in Section 4.

For simplification during testing, cell entity data was defined using 4-bit signed values. These were used to test the propagation of the maximum subarray from cell P_1 shown in Figure 3, to cell $P_{k \times j}$. The implementation was tested using a Xilinx, XC2VP2-5FG456 Virtex-II FPGA and DS-KIT-2VP4LC development system that incorporated an LCD interface to verify our results.

4 Test-bench Simulations

Several test benches were written in VHDL to simulate the implementation of the maximum sum algorithm. Cell values v , and identities (i, j) , were defined as constants, the former comprised 4-bit, signed integers, the latter, as 4-bit unsigned integers. Propagation of the partial sum throughout an $N \times N$ array for various values of N , were tested. The results of these tests showed correct operation of the cell entity and update of the partial sum, s , within the matrix structure shown in Figure 3.

5 Results

The utilisation of FPGA resources, in addition to actual performance tests, given for various array sizes, is shown in Table 1. These results show a significant increase in on-chip resource in proportion to array size.

Table 1: FPGA resource utilisation and performance measures for various cell sizes.

No. of Cells	4	16	36	64
Adders 4-bit	4	16	36	64
Registers 1-bit	51	183	403	711
Registers 4-bit	24	96	216	384
Comparators 4-bit	8	32	72	128
Clock cycles (Actual)	7	11	15	19

However, the performance of the maximum subarray algorithm, in terms of the number of clock cycles required to estimate a centroid, was significantly lower compared to the serial processing configuration shown in Figure 2. For each configuration tested, the actual number of clock cycles given in Table 1 shows an addition four cycles on theoretical estimates. This was due to the initialisation required for loading the data values into each cell entity. Optimal processing time was achieved in $T_{(N+M-1)+4}$ clock periods.

An estimate of the minimum number clock cycles required for centroid calculations using the method shown in Figure 2 is given as, $T_{(N \times N)}$ clock periods for a square array. For example, an 8×8 array would require a minimum of 64 clock cycles to calculate a centroid estimate.

6 Future Work

The Shack-Hartmann wavefront sensor produces an image similar to that shown in Figure 4. Here, the estimation of centroid pairs is required, each pair over an ROI, or sub-aperture, of size $N \times M$. Extending the number sub-apertures to an $R \times S$ array, where each sub-aperture corresponds to a ROI ζ , is ideally suited to the array structure shown in Figure 3, i.e., application of the maximum sum algorithm.

Independent centroid estimates, shown for example as A and B in Figure 4, combine to form wavefront measurements that can be corrected as discussed in Section 2.1. As the number of sub-apertures, and corresponding pixel density (or CCD array area) increases, the ability of the Shack-Hartmann sensor

to determine higher order wavefront aberrations [4] is increased.

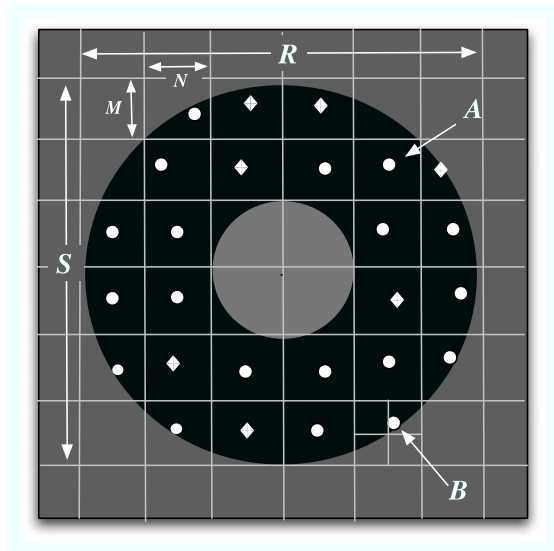


Figure 4: A Telescope Aperture showing Sub-aperture arrays for Centroiding using a Shack-Hartmann Image Sensor

However, given a fixed CCD size, if the number of sub-apertures increase, the number of pixels per sub-aperture will decrease. For example, Figure 4 shows an array of 6×6 sub-apertures over an active CCD array of dimensions, $R \times S$. Each sub-aperture is shown as an $N \times M$ pixel array. For optimal resolution, each sub-array should contain 4-pixels i.e., $N = M = 2$. This is known as a *quadcell*. If the number of sub-apertures were increased, as would be the case for example if a new configuration of CCD sensor and lenslets were fitted, reconfigurable hardware could be used to adjust the number of cell entities, incorporating the efficiency of the maximum sum algorithm.

Future work is required to extend this research for an application such as the Shack-Hartmann wavefront sensor.

7 Conclusion

In this paper we have given a brief outline of the maximum sum algorithm and shown how this algorithm can be implemented and tested using the hardware description language, VHDL.

Application of the maximum subarray algorithm for centroid estimation was proposed and a possible hardware application was outlined. Future work in the application of this research to a Shack-Hartmann wavefront sensor for use in adaptive optics is proposed.

8 Acknowledgements

We acknowledge the major contribution from Professor Tad Takaoka and Mr. Sung Bae, PhD candidate, from the department of Computer Science and Software Engineering at the University of Canterbury. Their support has provided much of the foundation for this research. In addition, their encouragement has been a significant factor in the success of this project.

References

- [1] R. Tyson, *Introduction to Adaptive Optics*. SPIE Press, 2000.
- [2] B. Pui, B. Hayes-Gill, M. Clark, M. Somekh, C. See, J. Pièri, S. Morgan, and A. Ng., "The design and characterisation of an optical VLSI processor for real time centroid detection," *Proceedings of SPIE*, no. 4408, pp. 73–80, 2001.
- [3] S. Thomas, "Optimized centroid computing in a shack-hartmann sensor," *Proceedings of SPIE*, no. 5490, pp. 1238–1246, 2004.
- [4] M. Roggermann and B. Welsh, *Imaging through turbulence*. CRC Press, 1996.
- [5] J. Bentley, *Programming Pearls*. Addison-Wesley, Inc., 2 ed., 2000.
- [6] S. E. Bae and T. Takaoka, "Parallel approaches to the maximum subarray problem," *Proc. of the seventh Japan-Korea Workshop on Algorithms and Computation*, pp. 94–104, 2003.
- [7] S. Bae and T. Takaoka, "Algorithms for the problem of k maximum sums and a VLSI algorithm for the k maximum subarrays problem," *2004 International Symposium on Parallel Architectures (ISPAN'04)*, pp. 247–253, 2004.

A study on GPU implementation of March's regularization method for optical flow computation

Yoshiki Mizukami, Katsumi Tadamura

Yamaguchi University, Faculty of Engineering, Japan.

Email: mizu@yamaguchi-u.ac.jp

Abstract

In this study, March's regularization method was implemented on GPUs for the investigation of fast and dense displacement computation. In the simulations, a sequential set of computer graphics images was employed. The GPU implementation provided more than 13 times the execution speed of the traditional CPU implementation. The coarse-to-fine strategy further increased the speed, making it 6 times faster. Compared with Horn and Schunck's method, which is one of the most representative approaches, March's method could compute the displacement more accurately and be accelerated more drastically by implementing it on GPUs. The GPU implementation of March's method realized the processing time of several frames per second, possibly making it useful for practical applications such as motion analysis and security systems.

Keywords: optical flow, GPU, regularization, March's method, Horn and Schunck's method

1 Introduction

The computation of two-dimensional displacement on sequential images, *optical flow*, is one of the main issues in image processing and have been employed in broad applications. Barron et al. surveyed several techniques in 1994 [1], and McCane et al. reported further benchmarking results in 2001 [2]. Among the many previously-proposed methods, Horn and Schunck's regularization method is one of the most representative approaches [3] and was deeply discussed in these survey studies.

March proposed a regularization method for binocular stereopsis in 1988 [4], which is applicable to the computation of optical flow [5] or the pattern recognition problem [6, 7]. March's method has a strong relationship with Horn and Schunck's method in that these two methods are based on the regularization theory [8] and consist of parallel computation. The same constraint of the departure from smoothness in computed displacement was used as a regularization term in these two methods, while the different corresponding term were used for measuring the error between intensity images. Horn and Schunck linearized the corresponding error by eliminating the second and higher order factors after applying Taylor expansion. On the other hand, March proposed a non-linear model, keeping the higher order factors. A comparison study in 2001 [9] conducted several simulations mainly based on simple synthesized sequential images and

revealed that March's method gave more accurate displacement with a lower number of iterations, but the computational cost per iteration was more expensive than that of Horn and Schunck's method.

For practical applications such as motion analysis and security systems, the computational time for computing displacement is essential. Until now, many researchers have attempted to develop specific hardware for computing displacement between images. Conventional studies employed analog VLSIs or FPGAs [10, 11]. In recent years the performance of Graphics Processing Unit (GPU) have been dramatically improved and have attracted much attention as a general-purpose parallel computing architecture [12]. For instance, in the field of computer vision and pattern recognition, Yang et al. proposed a real-time GPU implementation for computing the depth in binocular stereopsis [13], and Fung et al. [14] proposed a usage of multiple graphics cards. Deformable pattern recognition has also been implemented on GPUs [15]. In 2004, Strzodka [16] discussed the GPU implementation of an image registration method proposed by Clarenz et al. in 2002 [17], which has similarities with March's regularization method. Several remarks relating to the GPU implementation of Horn and Schunck's method have been made (e.g. [18, 19]).

The coarse-to-fine search strategy also should be very effective for reducing the computational time. Actually Yokoya [5] applied a coarse-to-fine strat-

egy to a method similar to March's method on the traditional CPU implementation. However his study did not clarify how much the coarse-to-fine strategy can reduce the computational time.

This study has three main purposes. The first purpose is to implement March's method on the GPU architecture and clarify its performance. The second purpose is to compare March's method with Horn and Schunck's method from the viewpoint of the GPU implementation. The third purpose is to investigate how much the coarse-to-fine strategy can reduce the computational time of March's method. Several simulations are conducted based on sequential computer graphics images, and the obtained results will be discussed in terms of the accuracy and computational time. Horn and Schunck's method is also studied for comparison.

Hereinafter, Section 2 briefly explains March's method, Horn and Schunck's method, the coarse-to-fine strategy and the GPU implementation. Section 3 shows the computational simulation and discusses the results. Section 4 describes the conclusions.

2 Techniques

Let us give a brief explanation of Horn and Schunck's method and March's method [3, 4]. Figure 1 illustrates two images, $f(x, y)$ and $g(x, y)$, and displacement function (u, v) at the coordinate (x, y) on g . In the framework of the regularization theory, the displacement function $(u(x, y), v(x, y))$ can be given by minimizing the following functional $E(u, v)$,

$$E(u, v) = P(u, v) + \lambda S(u, v) \quad (1)$$

$$P(u, v) = \iint (f_x u + f_y v + f_t)^2 dx dy, \quad (2)$$

$$S(u, v) = \iint ((u_x^2 + u_y^2) + (v_x^2 + v_y^2)) dx dy, \quad (3)$$

where functional P is the corresponding error between two images, f and g , and functional S is a constraint for the departure from the smoothness on the computed displacement. The subscript denotes the partial differential operator.

On the other hand, March introduced the following formulation instead of Eq. (2).

$$P(u, v) = \iint (f(x + u, y + v) - g(x, y))^2 dx dy, \quad (4)$$

where it should be noted that the corresponding error is treated with non-linear representation, while Horn and Schunck linearized it in Eq.(2).

On the basis of calculus of variations, the following iterative equations for Horn and Schunck's method are derived,

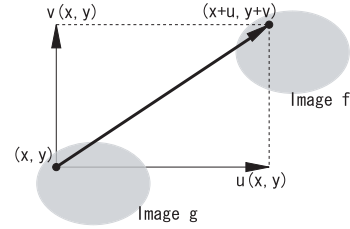


Figure 1: displacement function $(u(x,y), v(x,y))$

$$u^{[t+1]} = \bar{u}^{[t]} - \frac{f_x(f_x \bar{u}^{[t]} + f_y \bar{v}^{[t]} + f_t)}{\lambda + f_x^2 + f_y^2}, \quad (5)$$

$$v^{[t+1]} = \bar{v}^{[t]} - \frac{f_y(f_x \bar{u}^{[t]} + f_y \bar{v}^{[t]} + f_t)}{\lambda + f_x^2 + f_y^2}, \quad (6)$$

where (\bar{u}, \bar{v}) denotes the four-neighborhood average of (u, v) . On the other hand, March's method has the following iterative equations,

$$u^{[t+1]} = \bar{u}^{[t]} - \frac{1}{4\lambda} f_x (x + u^{[t]}, y + v^{[t]}) (f(x + u^{[t]}, y + v^{[t]}) - g(x, y)), \quad (7)$$

$$v^{[t+1]} = \bar{v}^{[t]} - \frac{1}{4\lambda} f_y (x + u^{[t]}, y + v^{[t]}) (f(x + u^{[t]}, y + v^{[t]}) - g(x, y)), \quad (8)$$

where it should be noted that the iterative equations require the subpixel value of f , f_x and f_y in the second terms of the right side, which are calculated by using the linear interpolation with the four surrounding pixel values. Instead of Eq.(7) and (8), this study employs the following equations, where (u, v) was substituted for (\bar{u}, \bar{v}) in the second term of the right side in order to improve the stability of the iterative computation [6, 7],

$$u^{[t+1]} = \bar{u}^{[t]} - \frac{1}{4\lambda} f_x (x + \bar{u}^{[t]}, y + \bar{v}^{[t]}) (f(x + \bar{u}^{[t]}, y + \bar{v}^{[t]}) - g(x, y)), \quad (9)$$

$$v^{[t+1]} = \bar{v}^{[t]} - \frac{1}{4\lambda} f_y (x + \bar{u}^{[t]}, y + \bar{v}^{[t]}) (f(x + \bar{u}^{[t]}, y + \bar{v}^{[t]}) - g(x, y)). \quad (10)$$

Let us explain the GPU implementation of Horn and Schunck's method and March's method. The above-mentioned iterative equations have the style of locally-parallel computation, so they seem to be very suited to the implementation on GPUs. Ordinary GPUs have two types of processors, that is, the vertex processor and the fragment processor. Since the fragment processor has multiple pipeline processing units and can store the computed results on the framebuffer in addition to retrieve data from the framebuffer [12], we decided to implement the iterative computation on the fragment processor. Since the 32-bit floating-point arithmetic on GPUs is provided through a set of OpenGL extensions, *GL_ARB_texture_float*, the degree of accuracy on GPUs compares favorably with CPUs.

Finally, let us outline the coarse-to-fine strategy. Before computing displacement between two images, the several pairs with lower resolution are generated from the original two images. The computation of displacement starts with the lowest resolutional pair, and the obtained low-resolutional displacement is utilized as the initial value for the finer pair. Eventually the finest resolutional displacement is computed with the original pair.

3 Simulations

Three types of simulations were conducted. In the first simulation, March's method and Horn and Schunck's method were implemented on CPUs, then the effects of the regularization parameter and the number of iterations on the computational accuracy were studied. In the second simulation, their iterative equations were implemented on GPUs, then their computational time and accuracy were investigated. In the third simulation, the coarse-to-fine strategy was adopted to the GPU implementation of displacement computation.

Our simulations employed a sequential set of computer graphics images, *Yosemite Fly-Through* [1]. There were 15 images and each image had 256 gray-scaled 315×252 pixels. The middle and next frames of the sequence were used as image f and g . As a preprocessing, these images are averaged with the equally-weighted filter of 3×3 pixels. The main specs of our computational environment are CPU Pentium 4 (3.2GHz), 2GB main memory, and the graphics card (NVIDIA GeForce 7800GT). Microsoft Visual Studio .NET 2003 and Cg Toolkit 1.4.1 were utilized for the programming environment. In this study, the root-mean-square was employed as the computational error ϵ for evaluating the computed displacement,

$$\epsilon = \frac{\sum_{x=X_b, y=Y_b}^{X-X_b, Y-Y_b} \sqrt{(u-u')^2 + (v-v')^2}}{(X-2X_b)(Y-2Y_b)}, \quad (11)$$

where X and Y are the width and height of the images, respectively, and $(u'(x, y), v'(x, y))$ is the correct value of horizontal and vertical displacement at the coordinate (x, y) on g . This equation takes no account of the region less X_b and Y_b -pixels apart from the border in horizontal and vertical direction, respectively. In this simulation, both X_b and Y_b were set to 10.

3.1 CPU implementation

Figure 2(a) and (b) show image $f(x, y)$ and $g(x, y)$, Fig.2(c) illustrates the correct displacement field,

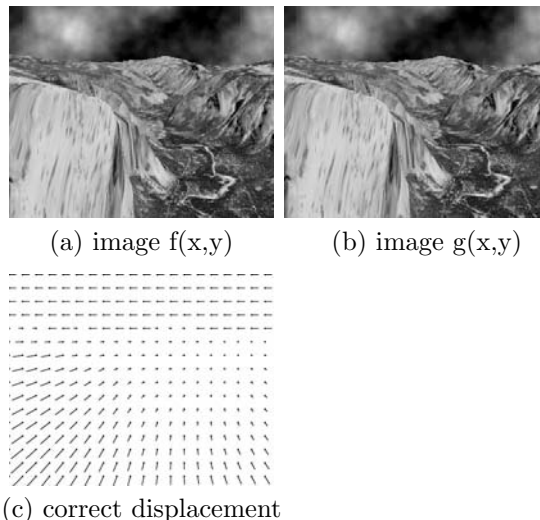


Figure 2: Yosemite-Fly and its correct displacement.

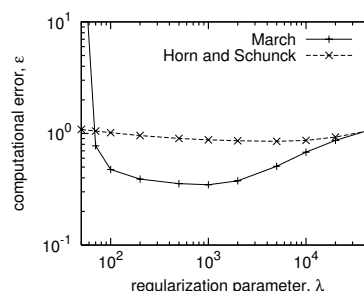
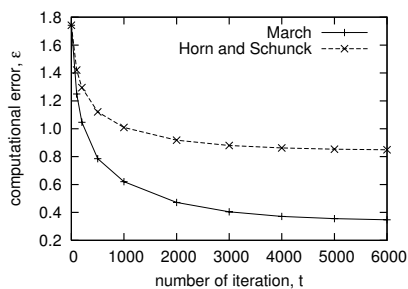


Figure 3: The influence of λ on computational error.

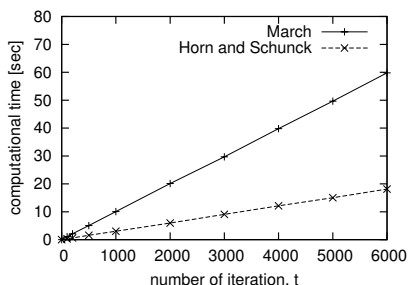
where the length of the subsampled displacement is quadrupled for visibility. Figure 3 shows the relationship between the regularization parameter λ and the computational error ϵ , where the number of iterations was set to 6,000. The figure shows that March's method gives smaller error in the broad range of the regularization parameter. The smallest error of March's method was 0.35 pixels at $\lambda = 1,000$, while that of Horn and Schunck's method was 0.85 pixels at $\lambda = 5,000$. The initial error was 1.742 pixels.

Next, let us discuss the number of iterations. Figure 4(a) and (b) illustrate the influence of the iteration number on the computational error and time, respectively, where the values of λ were set to 1,000 and 5,000 which gave the lowest error with each method in the above-mentioned simulation. The first figure shows that the errors of both methods decreased rapidly until around $t = 3,000$, when the computational error of March's method and Horn and Schunck's method are 0.40 and 0.88 pixels, respectively. Please note that March's method gave smaller error less than half of Horn and Schunck's method.

Figure 5(a) and (b) show the displacement fields computed with each method. In order to make clearly understandable, the error vector fields con-



(a) computational error



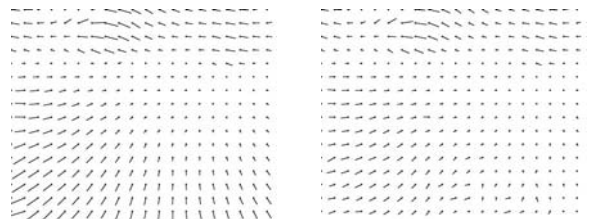
(b) computational time

Figure 4: results on CPU implementation.

tained in each result are illustrated in Fig. 5(c) and (d). Figure 5(c) indicates that March's method gave adequate displacement in mostly the regions of mountain surface and valley, but the wrong vertical displacement in the background sky region due to the changes in shape and intensity of clouds. Please note that the correct displacement of clouds is horizontal and constant as shown in Fig. 2(c). On the other hand, Fig. 5(d) indicates that Horn and Schunck's method gave inadequate displacement field in the broad region of mountain surface and valley. In the background sky region, Horn and Schunck's method gave the same or worse result than March's method.

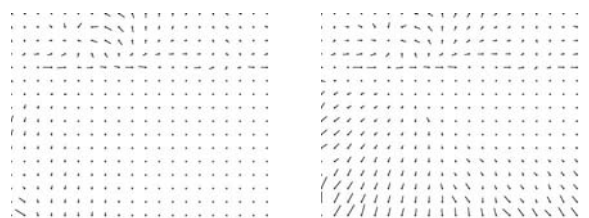
Let us discuss the computational time based on Fig. 4(b). The computational times required for the iteration of 3,000 steps were 29.71 and 9.06 sec in March's method and Horn and Schunck's method, respectively. It should be noted that March's method needed the computational time longer than 3.3 times of Horn and Schunck's method per iteration. The reason is that March's method requires the intensity and derivative at the subpixel coordinate in the iterative equations. In order to obtain these values, two-dimensional linear interpolation is applied based on four values of intensity, and derivatives at the neighboring pixel coordinates have to be retrieved. These drawbacks of the frequent access to the memory and the complicate computation result in requiring more computational time than Horn and Schunck's method.

As a consequence, it was indicated that March's method could give more accurate displacement



(a) result (March's)

(b) result (Horn et al.'s)



(c) error (March's)

(d) error (Horn et al.'s)

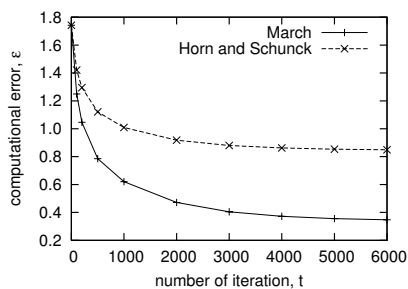
Figure 5: computed displacement and error

than Horn and Schunck's method, that March's method required longer computational time than Horn and Schunck's method per iteration, and that the implementations of these methods on CPU are too slow for practical applications which need fast processing.

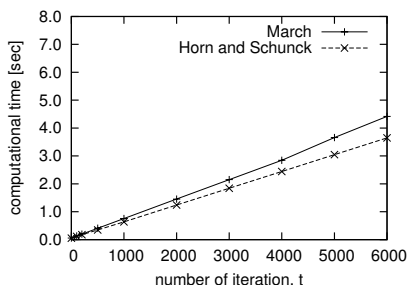
3.2 GPU implementation

Let us discuss the implementation of March's method and Horn and Schunck's method on GPUs. Figure 6(a) and (b) illustrate the influence of the iteration number on computational error and time, respectively. After the iteration of 3,000 steps, the computational error of these methods became 0.40 and 0.88 pixels, respectively, resulting in the same accuracy as the CPU implementation. On the other hands, the computational times were 2.15 and 1.84 sec. Please note that the GPU implementation could reduce the computational time of March's method by 13.8 times compared with the CPU implementation. The reason for the remarkable time reduction in March's method seems to be that the characteristics of GPU, that is, the multiple pipeline processing units and fast access to the framebuffer, could relief the above-mentioned drawbacks. Considering that the ratio between the computational times of two methods was 1.2, March's method is no longer much more expensive compared with Horn and Schunck's method.

The result of the GPU implementation indicated that March's method and Horn and Schunck's method were speeded up by 13.8 and 4.9 times, respectively, and that the computational time required for both methods was a few seconds.



(a) computational error



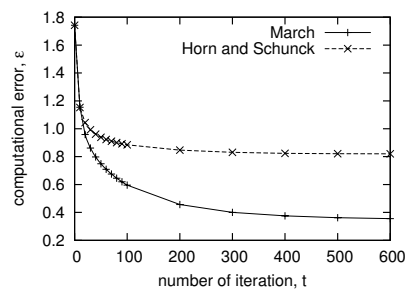
(b) computational time

Figure 6: result on GPU implementation

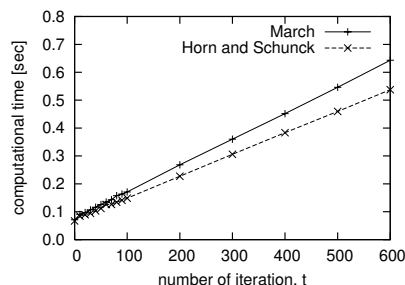
3.3 Multiscale GPU implementation

Next, let us discuss the adoption of the coarse-to-fine strategy to the GPU implementation. In this study, three different resolutional sets were utilized, that is, 316×252 , 158×126 and 79×63 . The intensity on the lower image was assigned with the averaged intensity of four corresponding pixel intensities on the finer images. As a result of the exploratory simulation, two sets of the regularization parameter, namely, $\{500, 500, 100\}$ and $\{10000, 20, 1\}$, were utilized for March's method and Horn and Schunck's method, respectively.

Figures 7(a) and (b) illustrate the influence of the iteration number on the computational error and time, respectively. The same iteration number was applied to all resolutional stages. Figure 7(a) shows that the iteration of 300 in March's method could provide the same computational error of 0.40 pixels as that of the iteration of 3,000 without using the coarse-to-fine strategy. Figure 7(b) indicates that the computational time was 0.36 sec and that the coarse-to-fine strategy could reduce the computational time to 1/6. On the other hand, in the case of Horn and Schunck's method, the iteration of 100 could provide the almost same computational error of 0.88 pixels as that of the iteration of 3,000 without using the coarse-to-fine strategy, where the computational time was only 0.15 sec. However, this small computational time does not mean that Horn and Schunck's method is more suitable to the GPU implementation with the coarse-to-fine strategy than March's method, since March's method could provide a smaller computational error of 0.68 pixels with



(a) computational error



(b) computational time

Figure 7: multiscale GPU implementation

the iteration of 70 steps and it only required the time of 0.14 sec as shown in Fig. 7(b).

The obtained results indicate that the implementation of March's method on GPUs is very effective for reducing the computational time, and that the adoption of the coarse-to-fine strategy to the GPU implementation makes possible for March's method to compute displacement between two images only in several hundred milliseconds.

4 Conclusion

In this study, March's regularization method was implemented on GPUs for establishing fast and dense displacement computation. In the simulations, a sequential set of computer graphics images, *Yosemite Fly-Through*, was employed. The GPU implementation could provide an execution speed more than 13 times greater than the traditional CPU implementation. The coarse-to-fine strategy could further increase the speed by 6 times. Compared with Horn and Schunck's method, which is one of the most representative approaches, March's method could compute the displacement more accurately and be accelerated more drastically by implementing it on GPUs. The GPU implementation of March's method realized the processing time of several frames per second which may make it useful for practical applications such as motion analysis and security systems.

There is a trade-off between computational time and the accuracy. In the situation which requires less accuracy, the computational time can be eliminated by reducing the number of iterations. Our

future work includes the extension to color images, further investigation with newer GPUs, the detection of the displacement discontinuities and the comparative study between March's method and Clarenz et al.'s method.

5 Acknowledgements

We express our gratitude to L. Quam et al. in SRI Lab., the authors of *Yosemite Fly-Through*. This study was partially supported by JSPS Grants-in-Aid for Science Research (16700208).

References

- [1] J. Barron, D. Fleet, and S. Beauchemin, "Performance of optical flow techniques," *Internat. J. Comput. Vision*, vol. 12, pp. 43–77, 1994.
- [2] B. McCane, K. Novins, D. Crannitch, and B. Galvin, "On benchmarking optical flow," *Computer Vision and Image Understanding*, vol. 84, no. 1, pp. 126–143, 2001.
- [3] B. Horn and B. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.
- [4] R. March, "Computation of stereo disparity using regularization," *Pattern Recognition Letters*, vol. 8, pp. 181–188, Mar. 1988.
- [5] N. Yokoya, "Dense matching of two views with large displacement," in *Proceedings of the 1st IEEE ICIP*, vol. I, pp. 213–217, Nov. 1994.
- [6] Y. Mizukami, K. Koga, and T. Torioka, "A handwritten character recognition system using hierarchical extraction of displacement," *IEICE*, vol. J77-D-II, pp. 2390–2393, Dec. 1994 (in Japanese).
- [7] Y. Mizukami and K. Koga, "A handwritten character recognition system using hierarchical displacement extraction algorithm," in *Proc. 13th Int. Conf. Pattern Recognition*, vol. 3, pp. 160–164, 1996.
- [8] T. Poggio, V. Torre, and C. Koch, "Computational vision and regularization theory," *Nature*, vol. 317, pp. 314–319, Sept. 1985.
- [9] Y. Mizukami, T. Sato, and K. Tanaka, "A comparison study for displacement computation," *Pattern Recognition Letters*, pp. 825–831, 2001.
- [10] R. Sarpeshkar, J. Kramer, G. Indiveri, and C. Koch, "Analog VLSI architectures for motion processing: From fundamental limits to system applications," *Proceedings of IEEE*, vol. X4, no. 7, 1996.
- [11] A. Zuloaga, J. Martin, and J. Ezquerria, "Hardware architecture for optical flow estimation in real time," *Proceedings of 1998 IEEE International Conference on Image Processing*, vol. 3, pp. 972–976, 1998.
- [12] J. D. Owens, D. Luebke, N. Govindaraju, M. Harris, J. Kruger, A. E. Lefohn, and T. J. Purcell, "A survey of general-purpose computation on graphics hardware," in *Eurographics 2005, State of the Art Reports*, pp. 21–51, Aug. 2005.
- [13] R. Yang and M. Pollefeys, "Multi-resolution real-time stereo on commodity graphics hardware," in *Proceedings of International Conference of Computer Vision and Pattern Recognition*, pp. 211–217, 2003.
- [14] J. Fung and S. Mann, "Using multiple graphics cards as a general purpose parallel computer : Applications to computer vision," in *Proceeding of International Conference of Pattern Recognition*, vol. 01, pp. 805–808, 2004.
- [15] Y. Mizukami and K. Tadamura, "GPU implementation of deformable pattern recognition using prototype-parallel displacement computation," in *Proc. DEFORM'06 - Workshop on Image Registration in Deformable Environments*, pp. 71–80, 2006.
- [16] R. Strzodka, M. Droske, and M. Rumpf, "Image registration by a regularized gradient flow - a streaming implementation in DX9 graphics hardware," *Computing*, vol. 73, no. 4, pp. 373–389, 2004.
- [17] U. Clarenz, M. Droske, and M. Rumpf, "Towards fast non-rigid registration," *Inverse Problems, Image Analysis and Medical Imaging, AMS Special Session Interaction of Inverse Problems and Image Analysis*, vol. 313, pp. 67–84, 2002.
- [18] P. Warden, "GPU optical flow," *Pete's GPU Notes*, 2005. http://petewarden.com/notes/archives/2005/05/gpu_optical_flow.html.
- [19] M. Gong, "A GPU-based algorithm for estimating 3D geometry and motion in near real-time," *Proceeding of the 3rd Canadian Conference on Computer and Robot Vision (CRV'06)*, pp. 10–, 2006.

Determination of Average Wind Velocity using Generalised SCIDAR

J.L. Mohr¹, R.A. Johnston², C.C. Worley¹, and P.L. Cottrell¹

¹Dept. Physics & Astronomy, University of Canterbury, Private Bag 4800, Christchurch 8020, New Zealand.

² Applied Research Associates NZ Ltd., PO Box 3894, Christchurch, New Zealand.

Email: j.mohr@phys.canterbury.ac.nz

Abstract

Distortion of images due to atmospheric turbulence is one of the major problems in astronomical imaging. To compensate for the induced aberrations in real-time it is vital to have an accurate model of the turbulence strength, $C_n^2(h)$, and the average wind velocity, $V(h)$.

At Mount John University Observatory a remote-sensing technique known as SCIDAR (SCIntillation Detection and Ranging) is used to determine the $C_n^2(h)$ and $V(h)$ profiles for the site. Preliminary results are presented for the wind velocity of near-ground turbulence.

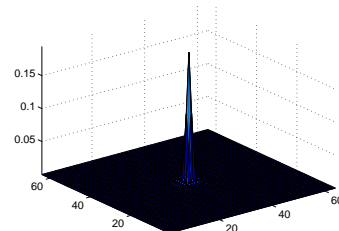
Keywords: atmospheric turbulence, site testing, SCIDAR, wind velocity

1 Introduction

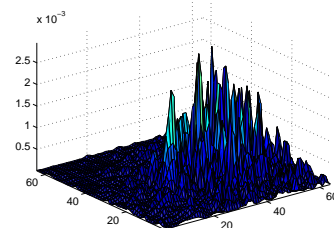
Distortion of images due to atmospheric turbulence is one of the major problems in astronomical imaging. The atmosphere is in a constant state of motion both spatially and temporally. This results in variations in refractive index, a function of humidity and temperature, inducing both phase and amplitude variations. The effect of the Earth's atmosphere on starlight can be visualised by thinking of the incoming wavefront as a flat sheet of paper. Atmospheric turbulence reduces the wavefront to a crumpled piece of paper and as a consequence distorts the resulting image. Initially the amplitude variations are too small to detect. With large propagation distances the strength of the amplitude fluctuations increases and can be seen as the *twinkling* of the stars. This effect is known as *scintillation*.

The effect of the atmospheric turbulence on the resolution of the image of a stellar object is shown in Figure 1. Ideally a diffraction limited image (Figure 1(a)) would be obtained. In reality the aberrations induced by the atmosphere make it difficult to distinguish between stellar objects as the resulting short exposure image has a speckled appearance (Figure 1(b)). The aim of image processing, post or real-time, is to take the crumpled wavefront and flatten it to approximate the incoming planar wavefront to obtain higher spatial resolution in the corrected images. Adaptive optics (AO) provides a real-time solution for the compensation of aberrations present in the incident wavefront. This is achieved by means of a closed-loop

system that utilises deformable optics. However for any AO system to be effective it is vital to have an understanding of the structure of atmospheric turbulence at a given site. A large number of AO design parameters rely on a prior knowledge of the turbulence structure and its characteristics.



(a)



(b)

Figure 1: The effect of atmospheric turbulence: (a) The ideal case or diffraction limited image of a star where no distortions are induced by the atmosphere. (b) The real situation where a speckle pattern is seen in the image due to the aberrations induced by the atmosphere.

One of the key parameters in the design of any AO system is the *Greenwood frequency*. The Green-

wood frequency, f_G , describes the rate at which the turbulence structure changes with time. It is defined as [1]

$$f_G = 2.31\lambda^{-6/5} \left[\sec \zeta \int_{\text{path}} C_n^2(h) V(h)^{5/3} dh \right]^{3/5}, \quad (1)$$

where λ is the wavelength, ζ is the zenith angle, $C_n^2(h)$ and $V(h)$ are the refractive index structure constant and the average wind velocity as a function of altitude h . $C_n^2(h)$ describes the vertical distribution of turbulence and along with $V(h)$ describe the properties of atmospheric turbulence. The Greenwood frequency, f_G , determines how quickly an AO system is required to respond to adequately compensate for the aberrations induced by the atmospheric turbulence. As such it is necessary to have accurate models of $C_n^2(h)$ and $V(h)$.

Although many techniques are available to provide measurements and estimates of $C_n^2(h)$ and $V(h)$, optical methods are preferred as they allow for measurements to be taken remotely. SCIDAR (SCIntillation Detection And Ranging) is such a technique and has been used at many different sites around the world [2, 3, 4, 5, 6, 7, 8].

In SCIDAR stellar scintillation patterns are measured at the telescope aperture. Any phase variations induced by atmospheric turbulence are very weak and must propagate large distances to produce measurable scintillation. As such any scintillation resulting from near-ground turbulence (NGT) will not be detected. A simple change of lens can virtually shift the measurement plane to below the telescope. This increases the propagation distance such that scintillation from NGT can now be measured. This version of SCIDAR is known as *generalised SCIDAR* [9].

This paper describes results from a bread-board based SCIDAR system, detailed in [10, 11]. $C_n^2(h)$ and $V(h)$ profiles resulting from NGT are presented. Also discussed is the effect of NGT on f_G values.

2 Theory

SCIDAR measurements are typically taken using a binary star, as shown in Figure 2. Light from each star passes through the same region of a turbulent layer forming identical, but separated, scintillation patterns. The distance the two patterns are separated is directly proportional to the angular separation of the binary star, ϕ , and the height of the turbulent layer, h .

Estimation of $C_n^2(h)$ requires inversion of the following matrix equation

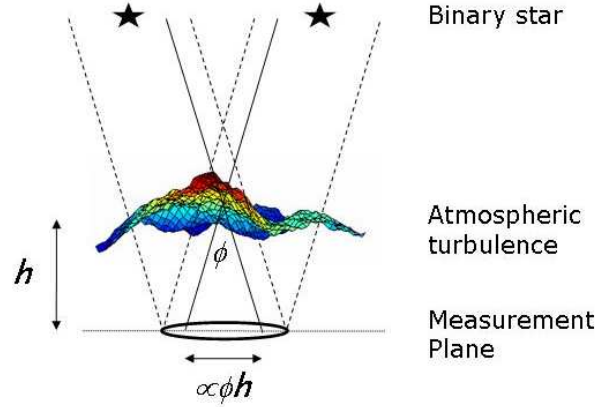


Figure 2: The concept of SCIDAR: Light from each star passes through the same point in the turbulent layer producing identical scintillation patterns separated by a distance proportional to the binary star separation ϕ and the height of the turbulent layer h .

$$S = T_S \times C_n^2(h), \quad (2)$$

where S is a 1D slice of the scintillation spatial covariance measured at the aperture of the telescope (noise removed) and T_S is a matrix of ideal covariances assuming Kolmogorov statistics for turbulence. The process is detailed, including full mathematical treatment, in [10, 12].

Spatial covariances, derived from the auto-correlation, consist of a group of triplets. Figure 3 shows a theoretical generalised SCIDAR spatial covariance with a virtual measurement plane of 2.5 km below the telescope. The central peak contains a contribution from all triplets with the strongest triplet resulting from NGT. The weaker triplet pattern is formed by turbulence located at an altitude of 10 km above the telescope. The distance between the primary and secondary peaks of each triplet set is proportional to the separation of the binary star and the sum of the distance of the telescope to the measurement plane and the turbulent layer.

Turbulent layers at different altitudes can be clearly identified from the spatial covariance. From Figure 3 it is not possible to determine how much of the NGT is associated with the dome and telescope or how much is outside or above the dome. As each NGT layer moves at a different velocity the identification of contributions from the dome and outside the dome can be seen in temporal covariances.

Temporal covariances, derived from the cross-correlation of two frames taken at a time dt apart, are also comprised of a group of triplets. However, unlike the spatial covariance, the local origin of each triplet is shifted by an amount proportional to the mean velocity $\langle V(h) \rangle$. Figure 4 shows the

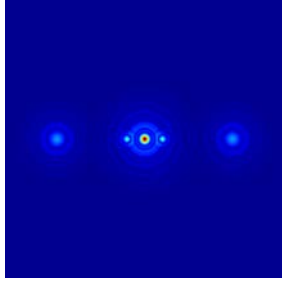


Figure 3: Spatial covariance of scintillation: The distance between the central peak and the secondary peak is proportional to the binary star separation and the height of the turbulent layer. This shows layers near the ground and at 10 km above the telescope. Measurement plane is 2.5 km below the telescope.

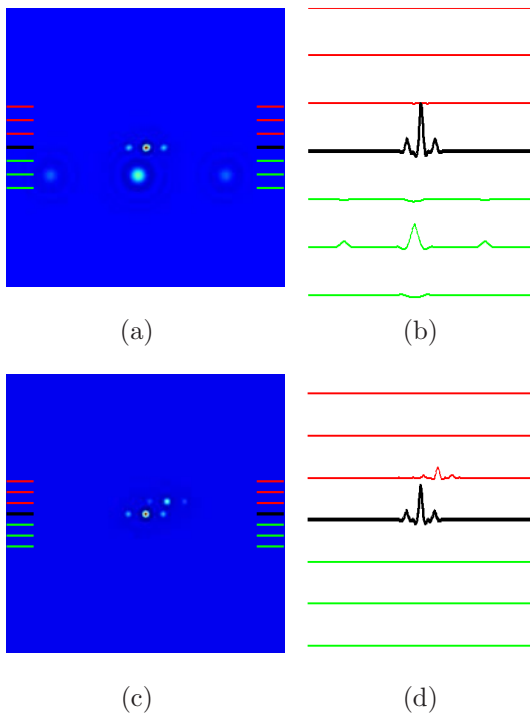


Figure 4: Temporal covariances of scintillation: The distance a triplet pattern has shifted from the image origin is proportional to the mean velocity of a given turbulent layer. (a) $dt = 10$ ms. ($\langle V(h)_{10\text{km}} \rangle = 20$ m/s) (b) Slice view of (a). (c) $dt = 90$ ms. ($\langle V(h)_{\text{BoundaryLayer}} \rangle = 2$ m/s) (d) Slice view of (c). Slice images are taken at positions indicated by the lines in (a) and (c).

theoretical temporal covariances and slices taken at positions indicated in the covariance images. The layers seen correspond to the layers found in Figure 3.

By measuring the displacement of the local origin of each triplet, ds , an average velocity $\langle V(h) \rangle$ given the known time difference between frames used, dt :

$$\langle V(h) \rangle = \frac{ds}{dt}. \quad (3)$$

Practically, determination of $\langle V(h) \rangle$ is problematic due to the weakened signal resulting from the shifted triplets and the resulting diminished signal-to-noise ratio. In addition a different value of dt is required to capture the motion of layers at different velocities.

If dt is sufficiently small the motion of a fast moving layer (such as a high altitude layer) can be readily detected (Figure 4(a) and (b)). However any movement in the NGT would not be detected due to the slow speeds of NGT. If dt is sufficiently large the motion of NGT layers can be seen (Figure 4(c) and (d)). In this case any motion from fast moving layers results in a shift greater than the sampled region. Therefore, to obtain information of the full $\langle V(h) \rangle$ profile it is necessary to take temporal covariances with at least two different dt values. Typically the time difference required to sample high altitude layer velocities, dt_{high} , is in the order of 10 - 20 ms [9]. The time difference required for NGT, dt_{NGT} , is significantly longer.

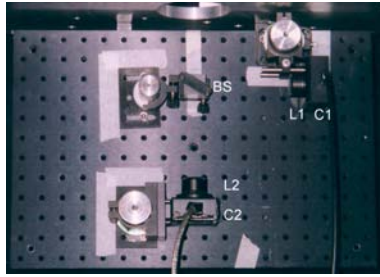
Although it is recognised that NGT is comprised of multiple layers [4], due to the slow moving nature of NGT typically little attention is given to NGT except to determine turbulence associated with the dome.

3 UC SCIDAR System

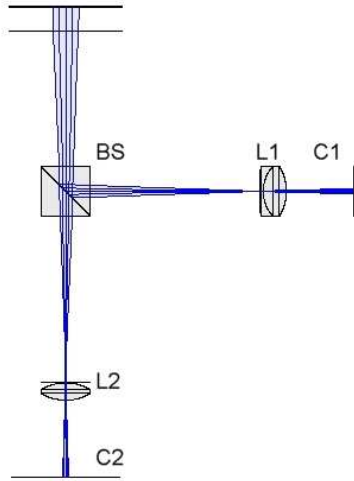
During 2005 SCIDAR measurements were taken at Mount John University Observatory (MJUO) using the UC SCIDAR system (described in [10, 11]) on the 1-m McLellan telescope. The UC SCIDAR system (shown in Figure 5) collects measurements from two different measurement planes simultaneously. SCIDAR data from a measurement plane at the telescope pupil is obtained using a f12.7mm achromat lens (L2). Generalised SCIDAR data from a measurement plane approximately 3.6km below the telescope is obtained using a f10mm achromat lens (L1). Both channels utilise a 640 x 480 CCD with 7.4 μm square pixels. SCIDAR data from a six minute period was collected in 20 blocks, each containing 250 frames from each channel.¹ Dark and sky frames were also captured.

Due to the combination of optics used, when the telescope is operating at a focal ratio of f/13.5, the sampled aperture has a radius of 63.5 pixels. To detect a moving layer the shifted triplet pattern in the temporal covariance should have a radial shift of no more than 50 pixels from the origin. A layer at 10-11 km above the observatory can be expected

¹Although the frame rate of the cameras were 30 Hz there is a period of time associated with writing the data block to disk. Hence to collect 5000 frames per camera (20 blocks) it takes approximately six minutes.



(a)



(b)

Figure 5: UC SCIDAR System: (a) physical layout; (b) optical layout.

to be moving at 15-35 m/s [1] depending on the wind model used. For a 50 pixel shift in the sampled aperture, where each pixel represents 0.79 cm, to detect a layer moving at 20 m/s the maximum time difference between two frames used in cross-correlations, dt , would be 19.7 ms. The cameras used in the UC SCIDAR system have a maximum frame rate of 30 Hz ($dt = 33.3\text{ms}$). Hence the fastest velocity that can be detected within a 50 pixel shift is 11.8m/s. Velocities of NGT are slow and hence will be easily detected.

NGT is typically not seen in data taken at the telescope aperture. As such only results from the generalised SCIDAR channel will be discussed.

4 Results and Discussion

A sample spatial covariance from the generalised channel, calculated from a six minute run (corresponding to 5000 frames), is shown in Figure 6. As expected NGT produces a strong triplet pattern in the centre. Also present is a triplet pattern produced by a high altitude layer at approximately 10 km above the telescope. The data from Figure 6 was taken using the binary star α Centaurus (α Cen) with an exposure time of 1 ms. At the time the data was collected (June 13 2005) the binary star separation of α Cen was 10.4 arcseconds. The

magnitude difference between the two stars of α Cen is 1.36. The estimated $C_n^2(h)$ profile is shown in Figure 7.

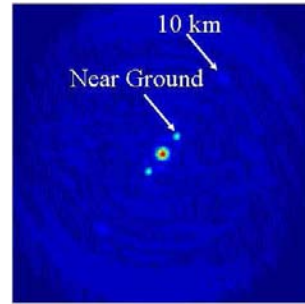


Figure 6: Spatial covariance from generalised SCIDAR data taken in June 2005.

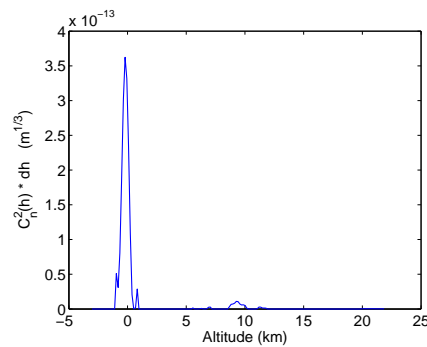


Figure 7: $C_n^2(h)$ profile from generalised SCIDAR data taken in June 2005.

Looking at the NGT the question becomes what is the required dt_{NGT} for movement of NGT layers to be detected. For the purpose of this discussion an analysis on four selected sequential blocks (1000 frames) is used. Figure 8 indicates the position and size of slices taken for analysis.

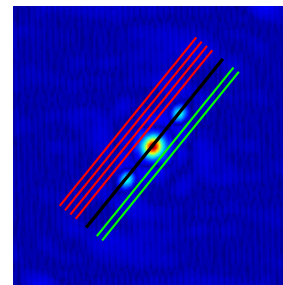


Figure 8: Position of slices taken.

Figure 9 shows temporal covariances and corresponding slice views calculated for various values of dt . Regardless of the dt value used it is clear that a significant portion of the NGT is associated with the dome. Any turbulence within the dome will not be moving and results in a triplet pattern that is located at the image origin (slice indicated by the black line). When dt is 33.3 ms (Figure 9(a)

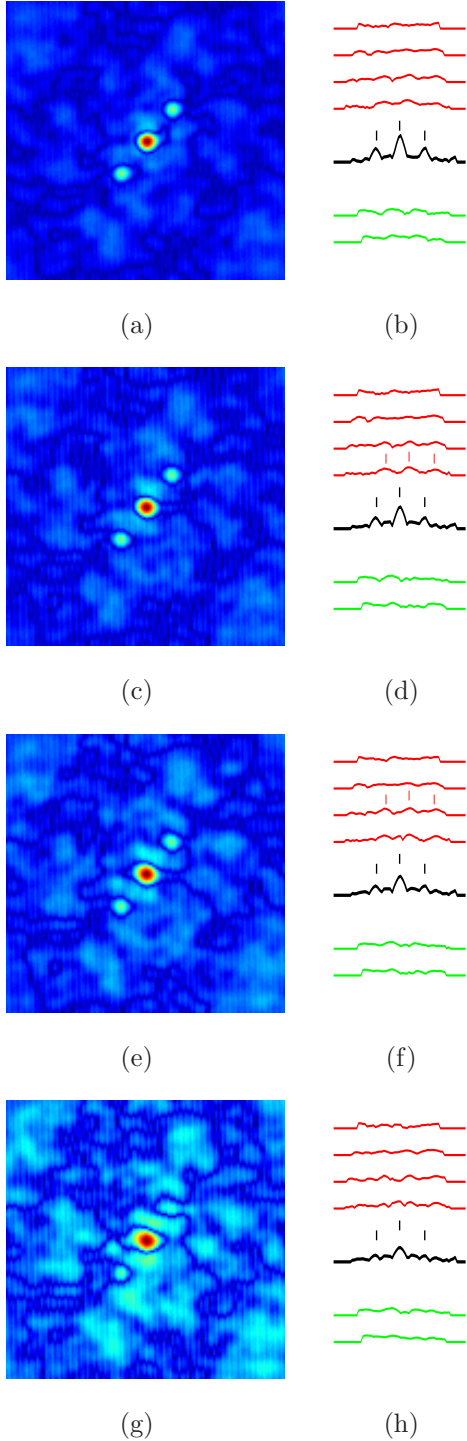


Figure 9: Temporal covariances and slice views from generalised SCIDAR data taken in June 2005. To bring peaks above the noise floor slices of $|\sqrt{\text{data}}|$ is shown. (a, b) $dt = 33.3$ ms (c, d) $dt = 99.9$ ms (e, f) $dt = 166.6$ ms (g, h) $dt = 333.3$ ms

and (b)) it is difficult to isolate any movement in layers. However when dt is 99.9 ms or 166.6 ms (Figures 9(c, d) and (e, f) respectively) a layer that is moving at approximately 0.5 m/s in an upward direction can be seen. This layer accounts for approximately 10% of the strength of the turbulence

associated with NGT. Increasing dt to 333.3 ms does not improve the ability to detect a layer just outside of the dome, but results in a de-correlation and increased noise levels. This suggests that to detect movement in NGT at MJUO dt_{NGT} should be limited to 100 - 160 ms.

It should be noted that the velocity direction in relation to magnetic north can not be identified as the orientation of the CCD sensor changes with respect to north as the telescope tracks the observed object. This will also result in a blurring of measured temporal covariance triplets.

In general for MJUO, NGT is comprised of at least two components: turbulence associated with the dome and turbulence associated with the boundary layer. As the turbulence resulting from the dome is not moving it would have no contribution to f_G (see equation 1). However the boundary layer turbulence, no matter how slowly it is moving, will contribute to f_G .

During the determination of $\langle V(h) \rangle$ each layer is treated as an infinitesimally thin layer. However from Figure 7 it can be seen that the strength of a turbulent layer is not fully represented by a single point. In calculating f_G the entire width of the estimated turbulent layer should be considered. The contribution of a turbulent layer to f_G can be found using

$$f_G(i) = 2.31\lambda^{-6/5} \left[\sec \zeta \langle V(i) \rangle^{5/3} \int_{i_0}^{i_1} C_n^2(h) dh \right]^{3/5} \quad (4)$$

where $\langle V(i) \rangle$ is constant for a given layer and i_0 and i_1 define the thickness of the layer.

Figure 10 shows how f_G for the boundary layer changes with wavelength given that imaging is at the zenith, the average wind velocity is 0.5 m/s and the integrated turbulence strength is $1.8 \times 10^{-13} \text{m}^{-2/3}$ for the boundary layer. When looking at the violet end of the visual light spectrum, f_G for the boundary layer is 1.24 Hz. Hence to fully compensate for aberrations induced by the boundary layer an AO system should be operating at a minimum of 5 Hz (four times f_G [1]).

In reality a bandwidth of 5 Hz is far from adequate to fully compensate for aberrations induced by NGT. The model of turbulence used in the above analysis assumes well-formed turbulent structure for a given layer. NGT is not well formed and hence the models used start to break-down for NGT. In addition the above calculation assumed a constant velocity throughout the thickness of the layer which is only known within the altitude resolution of the SCIDAR technique, which is dependent on the binary star separation (in

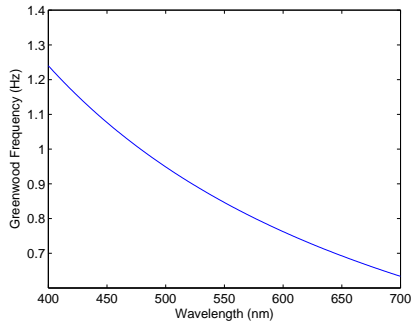


Figure 10: Variations in Greenwood frequency with respect to wavelength. This plot assumes that all turbulence is located in the boundary layer with a constant velocity of 0.5 m/s and an integrated $C_n^2(h)$ of $1.8 \times 10^{-13} m^{-2/3}$.

this case α Cen). All these factors introduce uncertainties in the determination of $C_n^2(h)$ and $V(h)$.

Another uncertainty will arise due to the fact that the wind velocity of NGT is not constant for long durations of time. Wind speeds can increase and decrease suddenly. The wind speeds determined above are an average over a very short period of time (approximately 60 s). The analysis should be extended for longer periods.

Although the above analysis provides a limitation to the values of dt_{NGT} , correlated patterns can also be seen in other regions of the data presented. These patterns may be the result of residual noise or telescope motion and warrant further investigation.

5 Conclusions

Due to the speed of turbulence at various altitudes the frame rate to capture movement of high altitude layers compared with that of the NGT varies considerably. To adequately detect velocities of NGT at MJUO one should limit values of dt_{NGT} to 100 - 160 ms. Values less than this may result in the lack of ability to detect movement in the NGT, whereas values greater than this can lead to a de-correlation.

For accurate determination of the effects of NGT on f_G , one should take an average wind speed over a long period, say 30 minutes. In addition the measurements should be correlated with the surface wind speeds measured at the site.

Further investigation into camera noise and telescope motion on temporal results should also be conducted.

References

- [1] R. K. Tyson and B. W. Frazier, *Field Guide to Adaptive Optics*, vol. FG03 of *SPIE Field Guides*. Bellingham, Washington: SPIE Press, 2004.
- [2] R. Avila *et al.*, “Whole atmospheric-turbulence profiling with generalized scidar,” *Appl. Opt.*, vol. 36, no. 30, pp. 7898–7905, 1997.
- [3] R. Avila *et al.*, “Turbulence Profiles with Generalized Scidar at San Pedro Martir Observatory and Isoplanatism Studies,” *PASP*, vol. 110, pp. 1106–1116, 1998.
- [4] R. Avila *et al.*, “Atmospheric turbulence and wind profiles monitoring with generalized scidar,” *A&A*, vol. 369, pp. 364–372, 2001.
- [5] J. J. Fuensalida *et al.*, “Cute-SCIDAR: An Automatically Controlled SCIDAR Instrument for the Jacobus Kapteyn Telescope,” *The ING Newsletter*, no. 8, pp. 15–18, 2004.
- [6] V. A. Klückers *et al.*, “Results from SCIDAR experiments,” in *Proceedings of SPIE Vol. 2828: Image Propagation Through the Atmosphere* (C. Dainty and L. R. Bissonnette, eds.), 1996.
- [7] J.-L. Prieur *et al.*, “SCIDAR measurements at Pic du Midi,” *A&A*, vol. 371, pp. 366–377, 2001.
- [8] A. R. Weiß *et al.*, “Simultaneous SCIDAR and adaptive optics measurements: results and applications,” in *Proceedings of SPIE Vol. 4538: Optics in Atmospheric Propagation and Adaptive Systems IV* (A. Kohnle, J. D. Gonglewski, and T. J. Schmutge, eds.), 2002.
- [9] V. A. Klückers *et al.*, “Profiling of atmospheric turbulence strength and velocity using a generalised SCIDAR technique,” *A&AS*, vol. 130, pp. 141–155, 1998.
- [10] R. A. Johnston *et al.*, “A bread-board SCIDAR system at Mount John,” in *Proceedings of Image and Vision Computing New Zealand 2004* (D. Pairman, H. North, and S. McNeill, eds.), 2004.
- [11] R. A. Johnston *et al.*, “Turbulence Monitoring at Mount John,” in *Proceedings of Image and Vision Computing New Zealand 2005* (B. McCane, ed.), 2005.
- [12] R. A. Johnston *et al.*, “Generalized scintillation detection and ranging results obtained by use of a modified inversion technique,” *Appl. Opt.*, vol. 41, no. 32, pp. 6768–6773, 2002.

Author Index

Adams, Nathan	67	Cottrell, P. L.	523
Ahnelt, P. K.	215	Cree, M. J.	233, 275, 487
Alfonso, Gastéllum	197		
Ali, Asad.....	85	De Silva, Liyanage C.	349
Allen, G. P.....	355	Delmas, Patrice.....	13, 103, 337, 461, 479
Allingham, Peter G.....	331	Dorrington, A. A.	233,487
Alwesh, Nawar S.	109, 245	Duan, Dandi	173
An, D.....	13		
Aouat, S.	431	Faggian, Nathan	73
Arakawa, Kenichi.....	473	Finn, A.....	161
Arnold, G.....	355	Flemmer, R. C.....	355
Aw, K.C.....	419	Flenley, J.....	355
		Fountain, D. W.	355
Bai, Li.....	361		
Bailey, D. G.....	31, 263, 467	Gao, Hongzhi	401
Bainbridge-Smith, A.....	281	Gao, Junbin.....	115
Barczak, Andre L. C.....	133, 395	Gibbins, D.	79, 161
Barrow, D. K.	311	Gilani, S. A. M.	85
Bashar, M. K.	137	Gilman, A.	31
Batatia, H.	325	Gimel'farb, Georgy.....	13, 461, 479, 493
Belliss, S. E.....	179	Gough, Christiaan A. D'H.....	299
Bennamoun, M.	257	Govignon, Quentin	239
Bhowan, Urvesh	425	Goyal, A.	343
Bickerton, Simon.....	239	Grant, Robert N.....	25
Billinghurst, Mark	299	Gray, D.....	79
Bischof, L.	227	Green, Richard.....	25, 67, 173, 203, 269, 293, 299, 311, 319, 367, 391, 401, 407
Blakeley, N. D.	343	Guo, Yi	115
Bloch, Isabelle	337		
Boles, Wageeh	499	Han, Jae-Sun	55
Bones, P. J.	215	Hashimoto, S.....	437
Brebner, D.	505	Hayes, Michael	287
Buckley, M.	227	Henriques, Alex.....	155
Bunnik, H. M. W.....	263	Hilsenstein, Volker	37, 331
Burling-Claridge, G. Robert	275	Hodgson, R. M.	355, 467
		Hoffman, A.	227
Carnegie, D. A.	233, 487	HONG, Yu-xuan.....	203
Chang, Eric Dahai.....	373	Huynh, C. P.....	293
Chen, Chia-Yen	91, 127	Hwang, Sun-Kyoo	443
Chen, Chi-Fa	127		
Chen, Yao-Hsin	19	Ikeda, H.....	437
Chu, Cheng-Tse	173	Irie, K.....	43
Clark, Adrian	367		
Collins, Michael J.....	499	Jackway, Paul	331
Conroy, R. M.....	233		

Janaqi, S.....	385	Mora, M.....	325
Johnson, M. J.	133, 395	Morris, A. B.	281
Johnston, R. A.	523	Morris, John	13, 239, 479, 493
Jorge, Márquez.....	197,337	Murshed, M.	413
Kanatani, Kenichi.....	7	Naeem, A.	449
Kang, Hyunchul	455	Newland, C.....	79
Kang, T.H.....	221	North, H. C.	179
Kelly, Brendon	407	Ny, Bunna	425
Kikkawa, M.	167	 	
Kim, H. C.	221	Oh, J. S.	221
Kim, M. G.	221	Ohnishi, N.	137
Kim, Whoi-Yul	55, 443	Osawa, Tatsuya	473
Kim, YongKyu	455	Owens, R.	257
Klette, Reinhard	1, 109	 	
Koo, J. M.....	221	Pairman, D.....	179
Kudo, H.....	137	Palmer, G. T.	245
Kwan, Paul W. H.....	115	Paplinski, Andrew	73
Kwon, Y.	281	Patrouix, Olivier.....	49
 		Pauwels, Eric J.	97
Lagerstrom, R.	227	Payne, A. D.....	233, 487
Langford, B. N.....	511	Penman, D. W.....	245
Le Ngoe-Thuy	305	Pereira, Nigel	349
Lee, Byeong Rae.....	455	Phang, S. S. S.	499
Lee, Gwang-Gook.....	55	Piccardi, Massimo.....	373
Lee, J. D.	221	Prakash, Surya	379
Li, J.	167	Prema, Vijay.....	251
Lim, Ee Hui	149, 191	Pridmore, T.	449
Lim Kah-Bin	305	Punchihewa, G. A. D.....	349, 467
Lin, Yizhe	13, 239	 	
Liu, J.	479	Raghavan, R.	419
Luthon, Franck.....	49	Rainbow, Amadeus.....	391
 		Ranguelova, Elena.....	97
Manning, S.....	167	Rayudu, R.	505
Marsland, S. R.....	355	Roberts, Gary.....	251
Matsumoto, T.....	137	Roberts, P.	161
Mawson, A. J.	263	Robles-Kelly, Antonio.....	379
McKinnon, A. E.	43	Rountree, R.....	505
McNeill, S. J.....	179	Rugis, John	185
Mian, A.....	257	Russell, S.	133
Millane, R. P.	215, 343	 	
Mills, S.	449	Saegusa, R.	437
Miskelly, Gordon	121	Sarjeant, S.	269
Mitra, A. K.	167	Schindler, Konrad	61
Miyagawa, Isao.....	473	Schmitt, Francis	337
Mizukami, Yoshiki.....	517	Schoo, Marcus	319
Mohr, J. L.	523	Schoonees, J. A.....	245
Montesinos, P.	385	Sherrah, Jamie.....	73

Shorin, Alexander	461	Wang, Hou-Wen	91
Sidibe, D.	385	Wang, King-Hang.....	19, 91
Sintorn, I.	227	Weddell, S. J.	511
Sorwar, G.....	413	Wenig, P.....	215
Souami, F.	431	Wojtas, D. H.....	215
Streeter, Lee	275	Woodhead, I. M.....	43
Sun, Hung-Min.....	19, 91	Woodward, Alexander.....	13
Suter, David	61, 149, 191	Worley, C. C.....	523
Swierkowski, L	161	Wu, B.	215
		Wuensche, Burkhard	143, 155, 251
Tadamura, Katsumi	517		
Takeuchi, Y.....	137	Xie, S.....	419
Tauber, C.....	325		
Tosas, Martin	361	Yasuno, Takayuki	473
Totozafiny, Théodore.....	49	Yoon, C. H.....	343
Turner, S.....	167	Yu, J. S.....	221
Uhlemann, Falk.....	209	Zang, Qi	1
Unsworth, K.	43	Zhang, Mengjie	425
		Zhang, Moqing	103
Valkenburg, Robert J.....	109, 245	Zhang, Rui.....	143
		Zhao, Yilan	109
Wagner, John	121	Zhou, Hang	61
Wakabayashi, Kaoru.....	473	Zhou, Zhen.....	49



ISBN 978-0-473-11792-4
ISBN 0-473-11792-4