Nasir Rajpoot, Abhir Bhalerao Department of Computer Science University of Warwick Coventry CV4 7AL United Kingdom

Proceedings of the British Machine Vision Conference 2007 (Warwick, Sept 2007)
Eds: N.M. Rajpoot, A.H. Bhalerao

ISBN Volumes 1 and 2: 978-0-902683-81-5
ISBN CD-ROM: 978-0-902683-82-2

British Library Cataloging in Publication Data
A catalogue record for this book is available from the British Library

Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may only be reproduced, stored or transmitted in any form or by means, with the prior permission in writing from the publishers, or in the reprographic reproduction in accordance with the terms of licenses issued by the Copyright Licensing Agency. Enquiries concerning reproduction outside those terms should be sent to the publishers.

©BMVA August 2007

The use of registered names or trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant laws and regulations and therefore free for general use.

The publisher makes no representation, express or implied, with regard to the accuracy of the information published in this book and cannot accept any legal responsibility for any errors of omission that may be made.

Hardcopy proceedings printed and bound in the United Kingdom by Warwick Print, University of Warwick, Coventry CV4 7AL.

Foreword

It is with great pleasure that we welcome you to BMVC 2007 at Warwick University. This year we received just over 300 submissions which is the second-highest number of submissions for BMVC, after an unpredictably large number of submissions last year. We believe that a growing number of international submissions to BMVC reflects its international prominence.

The task of reviewing was distributed over 87 experts (listed overleaf), each of whom on average reviewed 10 papers. The final selection took place at a meeting of the 23 Area Chairs on 18 June 2007 at the Computer Science Department, University of Warwick. A total of 114 papers were selected, 41 for oral presentation and 73 for poster presentation.

We are very pleased to have keynote addresses by Professor Hans Knutsson from the Linköping University in Sweden and Professor Mubarak Shah from the University of Central Florida. We are also delighted to have an invited tutorial on the emerging area of Visual SLAM by Dr Andrew Davison from the Imperial College, and Dr Andrew Calway and Dr Walterio Mayol-Cuevas from the University of Bristol.

We are grateful to Siemens and Warwick Warp for sponsoring the best security paper prize. The best science paper, the best poster, and the Sullivan thesis prizes are sponsored by the BMVA.

The organisation of the conference would not have been possible without the selfless help of many people whom we would like to thank. The reviewers and area chairs did a fantastic job of providing timely reviews and devoting much of their precious time to participate in the paper selection meeting. The CAWS team at Manchester University have been helpful in answering our queries related to the CAWS online system used for the conference. Manuel Trucco (BMVC'2006) was always very generous in providing tips and helpful advice on most matters regarding conference organisation. Majid Mirmehdi and Andrew Fitzgibbon (BMVA) offered almost instant help with general administrative as well as technical matters whenever asked. Catherine Pillet, our finance officer, has been invaluable in handling the registrations and delegate queries. Our thanks also to Jean Trevis of Warwick Conferences.

We would like to thank the staff and PhD students in the Computer Science Department at University of Warwick, especially those in the Signal and Image Processing and Medical Informatics and Medical Image Computing (MiMIC) research groups, for their help during the conference week. A special thanks to Muhammad Arif for double-checking the conference programme for us.

We hope that you find the conference informative and stimulating, and that you enjoy your stay at Warwick.

Nasir Rajpoot and Abhir Bhalerao
Warwick, July 2007

Area Chairs

Daniel Alexander
Mike Chantler
Ela Claridge
Timothy Cootes
E. Davies
Tom Drummond
Mark Everingham
Robert Fisher
Andrew Fitzgibbon
John Gilby
Shaogang Gong
Peter Hall
Richard Harvey
Graeme Jones
Chang-Tsun Li
Jun Liu
Jiri Matas
Majid Mirmehdi
Mark Nixon
Tomas Pajdla
Tony Pridmore
Simon Prince
Constantino Reyes-Aldasoro

Referees

Lourdes Agapito
Daniel C. Alexander
Ernesto L. Andrade
Muhammad Arif
Richard A. Baldock
Adrien E. Bartoli
Mohammed Bennamoun
Abhir H. Bhalerao
Horst Bischof
Edmond Boyer
Francois Bremond
Andrew D. Calway
Andrea Cavallaro
Dmitry Chetverikov
Bill Christmas
Javier Civera
Adrian F. Clark
John P. Collomosse
Timothy F. Cootes
Nicholas P. Costen
Antonio Criminisi
David Cristinacce
E. R. Davies
Tom W. Drummond
Mark R. Everingham
Paolo Favaro
Rob Fergus
Robert B. Fisher
Andrew W. Fitzgibbon
Andrea Fusiello
Aphrodite Galata
Andrea Giachetti
John H. Gilby
Oliver Grau
J.J. Guerrero
Peter M. Hall
Richard Harvey
Byung-Woo Hong
Roger J. Hubbold
Graeme A. Jones
Sohaib Khan
Andrew P. King
Bastian Leibe
Chang-Tsun Li
Xuelong Li
Marcus A. Magnor
Jiri G. Matas
Iain Matthews
Stephen J. Maybank
Walterio W. Mayol
Stephen J. McKenna
Ajmal S. Mian
Krystian Mikolajczyk
Majid Mirmehdi
JM M. Montiel
Mark S. Nixon
Tomas Pajdla
Yvan R. Petillot
Justus H. Piater
Stephen B. Pollard
Tony P. Pridmore
Simon J. Prince
Nasir Rajpoot
Constantino C. Reyes-Aldasoro
Neil M. Robertson
John Robinson
Antonio Robles-Kelly
Paul L. Rosin
Gerhard Roth
Ali Shahrokni
Jan P. Siebert
Fabrizio Smeraldi
William A. Smith
Jonathan Starck
Peter Sturm
Federico M. Sukno
Dacheng Tao
Neil Thacker
Tardi Tjahjadi
Bill Triggs
Carole J. Twining
Andrew M. Wallace
Jian-Gang Wang
Li Wang
Richard C. Wilson
Roland Wilson
David S. Young

Table of Contents

VOLUME I

Keynote Lecture

Manifolds and Images: Signal processing goes round the bend 1
Professor H. Knutsson (Medical Informatics Group, Department of Biomedical Engineering, Linköping University, Sweden)

Session 1: Tracking

Combining Local Appearance and Motion Cues for Occlusion Boundary Detection 2
A. N. Stein and M. Hebert (Carnegie Mellon University).

Motion Segmentation Using Inference in Dynamic Bayesian Networks 12
M. Toussaint (Technical University of Berlin), V. Willert (Honda Research Institute Europe), J. Eggert and E. Körner (Honda Research Institute Europe GmbH).

Layered Active Contours for Tracking 22
G. D. Pryor, P. Vela, T. u. Rehman and A. Tannenbaum (Georgia Institute of Technology).

Automatic Player Detection, Labeling and Tracking in Broadcast Soccer Video 32
J. Liu (Institute of Electronics, Chinese Academy of Sciences), X. Tong (Intel China Research Center), W. Li, T. Wang (Intel Corporation), Y. Zhang (Intel China Research Center), H. Wang (Institute of Electronics, Chinese Academy of Sciences), B. Yang, L. Sun and S. Yang (Tsinghua Univeristy).

Feature-Driven Direct Non-Rigid Image Registration 42
G.-B. Vincent (Lasmea), A. E. Bartoli (CNRS) and P. Sayd (CEA LIST).

Discovering Planes and Collapsing the State Space in Visual SLAM 52
A. P. Gee, D. Chekhlov, W. W. Mayol and A. D. Calway (University of Bristol).

Session 2: Surveillance/Registration

A Probabilistic Framework for Recognizing Similar Actions using Spatio-Temporal Features 62
A. Patron and I. Reid (University of Oxford).

Associating People Dropping off and Picking up Objects 72
D. Damen and D. Hogg (University of Leeds).

Spatial modelling of multi-layered LiDAR images using reversible jump MCMC 82
S. Hernandez-Marin, A. M. Wallace and G. J. Gibson (Heriot Watt University).

A New Framework for Automatic Registration of 2D/3D Texture Images . 92
A. S. El-Baz (University of Louisville) and G. Gimel'farb (University of Auckland,).

Fast Multigrid Optimal Mass Transport for Image Registration and Morphing 102
T. u. Rehman, G. Pryor and A. Tannenbaum (Georgia Institute of Technology).

Poster Session 1

Rotation, Rescaling and Occlusion Invariant Object Retrieval 112
M. Lecca and S. Messelodi (Fondazione Bruno Kessler).

Colour Constancy Based on Model Selection 122
C. Li and P. M. Hall (University of Bath).

Interest-point Based Face Recognition from Range Images 132
F. R. Al-Osaimi, M. Bennamoun and A. S. Mian (The University of Western Australia).

Towards Automated Visual Assessment of Progress in Construction Projects 142
T. C. Lukins (Heriot Watt University) and E. Trucco (University of Dundee).

Identifying Lens Distortions in Image Registration by Learning from Examples 152
B. Moeller and S. Posch (Martin-Luther-University Halle-Wittenberg).

Epsilon Stereo Pairs 162
J. Yu and Y. Ding (University of Delaware).

Learning object classes from structure 172
X. Bai, Y.-Z. Song and P. M. Hall (University of Bath).

Simple Representation and Approximate Search of Feature Vectors for Large-Scale Object Recognition 182
K. Kise, K. Noguchi and M. Iwamura (Osaka Prefecture University).

Enforcing 3D Constraints To Improve Object and Scene Recognition 192
R. Hewitt (University of California, San Diego), L. Goncalves (Evolution Robotics Retail) and M. E. Munich (Evolution Robotics).

An Automatic Framework for Figure-Ground Segmentation in Cluttered Backgrounds 202
L. Loss, G. Bebis, M. Nicolescu (University of Nevada, Reno) and A. N. Skurikhin (Los Alamos National Laboratory).

A framework for learning to recognize and segment object classes using weakly supervised training data 212
C. Pantofaru and M. Hebert (Carnegie Mellon University).

Image Enhancement Using Vector Quantisation Based Interpolation 222
W. P. Cockshott, S. L. Balasuriya, I. P. Gunawan and J. P. Siebert (University of Glasgow).

Camera calibration for miniature, low-cost, wide-angle imaging systems . . 232
O. Frank, R. Katz, C.-L. Tisse and H. Durrant-Whyte (University of Sydney).

Beyond Facial Expressions: Learning Human Emotion from Body Gestures 242
C. Shan, S. Gong and P. W. McOwan (Queen Mary, University of London).

The DIARETDB1 diabetic retinopathy database and evaluation protocol 252
T. Kauppi (Lappeenranta University of Technology), V. Kalesnykiene (University of Kuopio), J.-K. Kamarainen, L. Lensu (Lappeenranta University of Technology), I. Sorri, A. Raninen, R. Voutilainen (University of Kuopio), H. Uusitalo (University of Tampere), H. Kälviäinen (Lappeenranta University of Technology) and J. Pietilä (Perimetria Ltd.).

Hilbert-Huang Transform-based Local Regions Descriptors 262
D. Han, W. Li, W. Guo (Jilin University) and Z. Li (Shandong University of Technology).

Incremental LDA Learning by Combining Reconstructive and Discriminative Approaches 272
M. Uray (Graz University of Technology), D. Skocaj (University of Ljubljana, Faculty of Computer and Information Science), P. M. Roth, H. Bischof (Graz University of Technology) and A. Leonardis (University of Ljubljana, Faculty of Computer and Information Science).

High Accuracy Computation of Rank-Constrained Fundamental Matrix . . 282
Y. Sugaya (Toyohashi University of Technology) and K. Kanatani (Okayama University).

A single-perspective novel panoramic view from radially distorted non-central images 292
R. Molana and K. Daniilidis (University of Pennsylvania).

Robust Active Appearance Models with Iteratively Rescaled Kernels 302
M. G. Roberts, T. F. Cootes and J. E. Adams (The University of Manchester).

Unsupervised Learning of Shape Manifolds 312
N. Rajpoot (University of Warwick), M. Arif (Pakistan Institute of Engineering and Applied Sciences (PIEAS)) and A. H. Bhalerao (University of Warwick).

Finsler Level Set Segmentation for Imagery in Oriented Domains 322
V. Mohan, J. Melonakos (Georgia Institute of Technology), M. Niethammer, M. Kubicki (Harvard Medical School and Brigham and Women's Hospital) and A. Tannenbaum (Georgia Institute of Technology).

Automated Segmentation of Low-Light Level Imagery using Poisson MAP-MRF Labelling 331
H. Gribben, P. Miller, H. Wang (Queen's University Belfast) and M. Browne (Andor Technology).

Real-Time Humans Detection in Urban Scenes 341
J. BEGARD, N. ALLEZARD and P. SAYD (CEA LIST).

Class-Specific Binary Correlograms for Object Recognition 351
J. Amores (INRIA), N. Sebe (University of Amsterdam) and P. Radeva (Computer Vision Center).

Generalised Linear Pose Estimation 361
A. Ess, A. Neubeck and L. van Gool (ETH Zurich).

Overcoming Parallax and Sampling Density Issues in Image Mosaicing of Non-Planar Scenes 371
Z. QI and J. R. Cooperstock (McGill University).

Bayesian Surface Estimation from Multiple Cameras Using a Prior Based on the Visual Hull and its Application to Image Based Rendering 381
A. Mullins, A. Bowen, R. G. Wilson and N. Rajpoot (University of Warwick).

A multi-class SVM classifier for automatic hand washing quality assessment 389
D. F. Llorca (Alcala University), F. Vilarino, J. Zhou and G. Lacey (Trinity College Dublin).

Towards Real-Time Traffic Sign Recognition by Class-Specific Discriminative Features 399
A. Ruta (Brunel University, West London), Y. Li (Brunel University) and X. Liu (Brunel University, West London).

Unsupervised extraction of coherent regions for image based rendering . . . 409
J. Berent and P. L. Dragotti (Imperial College London).

Local Image Features for Shoeprint Image Retrieval 419
H. Su, D. Crookes, A. Bouridane and M. Gueham (Queen's University, Belfast).

Colour Transfer by Feature Based Histogram Registration 429
C. R. Senanayake and D. C. Alexander (University College London).

Super-resolution of faces using the epipolar constraint 439
R. Den Hollander, D.-J. De Lange and K. Schutte (TNO Defence, Security and Safety).

Fibre Centred Tensor Faces 449
B. P. Tiddeman, D. W. Hunter and Y. Meng (University of st andrews).

Binary Co-occurrences of Weak Descriptors 459
M. Winter and H. Bischof (Institute for Computer Graphics and Vision (ICG)).

Session 3: Biometrics

Capturing Correlations Among Facial Parts for Facial Expression Analysis 469
C. Shan, S. Gong and P. W. McOwan (Queen Mary, University of London).

Facial Emotions and Emotion Intensity Levels Classification and Classification Evaluation 479
M. Beszéde (Slovak University of Technology) and P. F. Culverhouse (University of Plymouth).

Segmented AAMs Improve Person-Indepedent Face Fitting 489
J. Peyras (Università Statale di Milano).

Gender Classification using Shape from Shading 499

J. Wu (the university of york, uk), W. A. Smith and E. R. Hancock (University of York).

Who are you? - real-time person identification 509

N. Apostoloff and A. Zisserman (University of Oxford).

Keynote Lecture

Recognizing Actions, Objects, and Actions as Objects 519

Professor Mubarak Shah, School of Electrical Engineering and Computer Science, University of Central Florida, USA

Session 4: Segmentation

A variational framework combining level-sets and thresholding. 520

S. Dambreville, A. Tannenbaum, A. Yezzi and M. Niethammer (Georgia Institute of Technology).

Automatic 3D Object Segmentation in Multiple Views using Volumetric Graph-Cuts 530

N. D. F. Campbell, G. Vogiatzis, C. Hernandez and R. Cipolla (University of Cambridge).

Zipfs Law in Image Coding Schemes 540

M. S. Crosier and L. D. Griffin (University College London).

Segmenting Highly Textured Nonstationary Background 550

D. M. Russell and S. Gong (Queen Mary, University of London).

Improving Spatial Support for Objects via Multiple Segmentations 560

T. Malisiewicz and A. A. Efros (Carnegie Mellon University).

Delving into the whorl of flower segmentation 570

M.-E. Nilsback and A. Zisserman (University of Oxford).

VOLUME II

Session 5: Image/Video Retrieval

Topology-Preserved Diffusion Distance for Histogram Comparison 580
W. Yan (Institute of Automation, Chinese Academy of Sciences), Q. Wang (Stanford University), Q. Liu, H. Lu and S. Ma (Institute of Automation, Chinese Academy of Sciences).

Distance-Free Image Retrieval Based on Stochastic Diffusion over Bipartite Graphs 590
C. Bauckhage (Deutsche Telekom).

Word Co-occurrence and Markov Random Fields for Improving Automatic Image Annotation 600
H. J. Escalante, M. Montes y Gómez and L. E. Sucar (Instituto Nacional de Astrofísica Optica y Electrónica (INAOE)).

Image Retrieval through Qualitative Representations over Semantic Features 610
Z. U. Qayyum and A. G. Cohn (University of Leeds).

Conditional Random Field for Natural Scene Categorization 620
Y. Wang and s. Gong (Queen Mary, University of London).

Poster Session 2

Evolutionary feature selection for probabilistic object recognition, novel object detection and object saliency estimation using GMMs 630
L. Trujillo (CICESE Ensenada,Mexico), G. Olague (CICESE), F. Fernandez (Universidad de Extremadura) and E. Lutton (INRIA).

Navier-Stokes formulation for modelling turbulent optical flow 640
A. Doshi and A. G. Bors (University of York).

All Pairs Shortest Path Formulation for Multiple Object Tracking with Application to Tennis Video Analysis 650
F. Yan, B. Christmas and J. Kittler (University of Surrey).

Shape from Texture: Fast Estimation of Planar Surface Orientation via Fourier Analysis 660
F. Galasso and J. Lasenby (Dept. of Engineering, Signal Processing Lab., University of Cambridge).

MOTEXATION: Multiple Object Tracking with Expectation-Maximization Algorithm 670
Y. Jin and F. Mokhtarian (University of Surrey).

Multicues 3D Monocular Upper Body Tracking Using Constrained Belief Propagation 680
P. Noriega and O. Bernier (France Télécom R&D).

Robust Multi-View Change Detection 690
A. Lanza, L. Di Stefano (University of Bologna), J. Berclaz, F. Fleuret and P. Fua (Ecole Polytechnique Federale de Lausanne (EPFL)).

Higher-order Autoregressive Models for Dynamic Textures 700
M. Hyndman, A. Jepson and D. J. Fleet (University of Toronto).

Refining implicit function representations of 3-D scenes 710
M. Grum and A. G. Bors (University of York).

Structure from Motion via Two-State Pipeline of Extended Kalman Filters 720
B. Clipp, G. F. Welch, J.-M. Frahm and M. Pollefeys (The University of North Carolina at Chapel Hill).

Time Varying Volumetric Scene Reconstruction Using Scene Flow 730
T. M. A. Smith, D. Redmill, N. Canagarajah and D. Bull (University of Bristol).

Human Pose Extraction from Monocular Videos using Constrained Non-Rigid Factorization 740
A. Shaji, S. Chandran (Indian Institute of Technology, Bombay), B. Siddiquie (University of Maryland) and D. Suter (Monash University).

Fast Motion Estimation on Range Image Sequences acquired with a 3-D Camera 750
S. Matzka, Y. R. Petillot and A. M. Wallace (Heriot Watt University).

A Practical Approach for Super-Resolution using Photometric cue and Graph Cuts 760
S. Sharma and M. V. Joshi (Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar, India).

Multi-scale Adaptive Mask 3D Rigid Registration of Ultrasound and CT Images 770
Z. Zhijun, S. Jerome, M. K. Soo, B. Yan and C. K. Yeung (The Chinese University of Hong Kong).

Geometrical constraint based 3D reconstruction using implicit coplanarities 780
R. Furukawa (Hiroshima City University) and H. Kawasaki (Saitama University).

Tracking Using Online Feature Selection and a Local Generative Model . . 790
T. E. Woodley (University of Cambridge), B. Stenger (Toshiba Research Europe) and R. Cipolla (University of Cambridge).

An Evaluation of Shape Descriptors for Image Retrieval in Human Pose Estimation 800
P. A. Tresadern (University of Salford) and I. Reid (University of Oxford).

Layered image model using binary PCA transparency masks 810
Z. Zivkovic (University of Amsterdam).

Isolating Motion and Color in a Motion Blurred Image 820
A. Giusti and V. Caglioti (Politecnico di Milano).

Improved Face Model Fitting on Video Sequences 830
x. Liu, F. Wheeler and P. Tu (GE Global Research).

Fitting Surface of Free Form Objects using Optimized NURBS Patches Network with Evolutionary Strategies (mu + lambda) - ES 840
J. W. Branch (Universidad Nacional de Colombia-Sede Medellin), F. Prieto (Universidad Nacional de Colombia - Sede Manizales) and P. Boulanger (University of Alberta).

Fully Automated Laser Range Calibration 850
M. Antone and Y. Friedman (BAE Systems AIT).

A Combined RANSAC-Hough Transform Algorithm for Fundamental Matrix Estimation 860
R. Den Hollander (TNO Defence, Security and Safety) and A. Hanjalic (Delft University of Technology).

Automatic Identification of Morphometric Landmarks in Digital Images . . 870
S. Palaniswamy, N. Thacker and C. P. Klingenberg (The University of Manchester).

Boosted Regression Active Shape Models 880
D. Cristinacce and T. F. Cootes (The University of Manchester).

Tracking Through Clutter Using Graph Cuts 890
J. Malcolm, Y. Rathi and A. Tannenbaum (Georgia Institute of Technology).

Shape Recovery Using Stochastic Heat Flow 900
V. P. Namboodiri and S. Chaudhuri (Indian Institute of Technology, Bombay).

Integrating Stereo with Shape-from-Shading derived Orientation Information 910
T. S. F. Haines and R. C. Wilson (University of York).

Active Segmentation and Adaptive Tracking Using Level Sets 920
Z. Chen and A. M. Wallace (Heriot Watt University).

Non-Gibbsian Markov random field models for contextual labelling of structured scenes 930
D. Heesch (Imperial College, South Kensington Campus, London SW7 2AZ) and M. Petrou (Imperial College London).

Denoising Manifold and Non-Manifold Point Clouds 940
R. Unnikrishnan and M. Hebert (Carnegie Mellon University).

Generic and Real-Time Structure from Motion 950
E. Mouragnon (CEA), M. Lhuillier (Universite Blaise Pascal), M. Dhome (CNRS/Université Blaise Pascal), F. Dekeyser and P. Sayd (CEA).

A phase field model incorporating generic and specific prior knowledge applied to road network extraction from VHR satellite images 960
T. Peng (Chinese Academy of Sciences), I. H. Jermyn (INRIA), V. Prinet (Chinese Academy of Sciences), J. Zerubia (INRIA) and B. Hu (Chinese Academy of Sciences).

Managing Particle Spread via Hybrid Particle Filter/Kernel Mean Shift Tracking 970
T. P. Pridmore, A. Naeem and S. Mills (Nottingham University).

Batch Algorithm with Additional Shape Constraints for Non-Rigid Factorization 980
Y. R. Loke and S. Ranganath (National University of Singapore).

Session 6: Object Recognition

Retina Sampling Feature Detection and Saccades; A Statistical Perspective. 990
N. Thacker (The University of Manchester) and C. Leek (University of Wales Bangor).

Unsupervised Category Discovery in Images Using Sparse Neural Coding . 1000
S. Waydo and C. Koch (California Institute of Technology).

Improvement of Retrieval Speed and Required Amount of Memory for Geometric Hashing by Combining Local Invariants 1010
M. Iwamura, T. Nakai and K. Kise (Osaka Prefecture University).

Object Detection Using Shape Codebook 1020
X. Yu and L. Yi (University of Maryland).

Generic Object Recognition via Shock Patch Fragments 1030
O. C. Ozcanli and B. B. Kimia (Brown University).

Session 7: Shape Modelling and Analysis

Implicit Active Model using Radial Basis Function Interpolated Level Sets 1040
X. Xie and M. Mirmehdi (University of Bristol).

Using Priors for Improving Generalization in Non-Rigid Structure-from-Motion 1050
S. I. Olsen (University of Copenhagen) and A. E. Bartoli (CNRS).

Automated Analysis of Deformable Structure in Groups of Images 1060
V. S. Petrovic, T. F. Cootes, A. M. Mills, C. J. Twining and C. J. Taylor (The University of Manchester).

Distribution-based Level Set Segmentation for Brain MR Images 1070
J. Liu, D. Chelberg (Ohio University), C. Smith and H. Chebrolu (University of Kentucky).

Sparse MRF Appearance Models for Fast Anatomical Structure Localisation 1080
R. Donner (Graz University of Technology), B. Micusik (Vienna University of Technology), G. Langs and H. Bischof (Graz University of Technology).

Session 8: Video Analysis

Indoor Place Recognition using Online Independent Support Vector Machines 1090
F. Orabona (University of Genoa), C. Castellini (LIRA-Lab, University of Genova, Italy), B. Caputo (IDIAP), J. Luo (IDIAP Research Institute) and G. Sandini (Italian Institute of Technology).

Video-rate recognition and localization for wearable cameras 1100
R. O. Castle, D. J. Gawley, G. Klein and D. W. Murray (University of Oxford).

Probabilistic egomotion from a statistical framework 1110
H. Shah and A. Lakshmikumar (Sarnoff Corporation).

On New View Synthesis Using Multiview Stereo 1120
O. J. Woodford, I. Reid (University of Oxford), P. Torr (Oxford Brookes University) and A. W. Fitzgibbon (Microsoft Research).

Manifolds and Images: Signal processing goes round the bend

Professor H. Knutsson (Medical Informatics Group, Department of Biomedical Engineering, Linköping University, Sweden)

Abstract

Signal and image processing aims at solving real world practical problems dealing with uncertain noisy signals generated by physical sensors. Traditional signal processing is based on a well-established theory developed for scalar signals in a statistical framework. A substantial part of the theory involves convolution operations. Smoothing and edge detection are simple examples of standard procedures in traditional image processing. Over the years there has been a need to generalize these methods, originally developed for monochrome 2-D images, to multi-dimensional scalar, vector and tensor fields.

Today some researchers are beginning to realize that the developed theoretical framework is still too narrow to handle a number of important problems. The fundamental limitation is that the theory is based on a Cartesian assumption that may not be valid. There are two typical situations where a Cartesian assumption can lead to severe errors: 1. The data sampling grid is non-Cartesian in space-time, e.g. ultrasound images or samples taken from curved surfaces. 2. The appropriate space for representing the sample values is a curved subspace (i.e. not a vector space) embedded in the measured parameter space, e.g. cyclic features, like hue and phase, or orientation.

The natural mathematical tools here is differential geometry, tensor analysis and the theory of smooth Riemannian manifolds. A manifold is a generalization of a surface in R3 where the properties can be defined without reference to an ambient space. Recent work reflect a growing interest in this topic, but a unified framework is still missing. We present a number of examples from our work, ranging from non-rigid registration (the Morphon) to manifold learning (the Sample LogMap), indicating the value of establishing the necessary steps for adapting traditional multi-dimensional signal processing methods to comprise general class of signals on manifolds, i.e why signal processing needs to "go round the bend".

Combining Local Appearance and Motion Cues for Occlusion Boundary Detection

Andrew Stein* and Martial Hebert
The Robotics Institute, Carnegie Mellon University
Pittsburgh, Pennsylvania, USA
anstein@cmu.edu

Abstract

Building on recent advances in the detection of appearance edges from multiple local cues, we present an approach for detecting occlusion boundaries which also incorporates local motion information. We argue that these boundaries have physical significance which makes them important for many high-level vision tasks and that motion offers a unique, often critical source of additional information for detecting them. We provide a new dataset of natural image sequences with labeled occlusion boundaries, on which we learn a classifier that leverages appearance cues along with motion estimates from either side of an edge. We demonstrate improved performance for pixelwise differentiation of occlusion boundaries from non-occluding edges by combining these weak local cues, as compared to using them separately. The results are suitable as improved input to subsequent mid- or high-level reasoning methods.

1 Introduction

Occlusion boundaries are a rich source of information in images. Not only do they provide boundary conditions for almost *any* process which reasons spatially within an image (*e.g.* optical flow, shape-from-X methods, feature extraction, filtering, *etc.*), but they also capture important perceptual information about the 3D scene [2]. Rather than being considered merely a nuisance to be "handled" or outliers to be avoided, as is often the case, these boundaries offer *opportunities* for segmentation and object discovery [3, 12, 14], and for reasoning about shape and structure [18].

Since occlusion boundaries correspond to locations where one object or surface is closer to the camera than another, we can exploit the resulting depth discontinuity as an indication of their existance. Noting that in many applications, video rather than single isolated images may be available, we can use local *motion* estimates as evidence of those depth discontinuities. In addition, most occlusion boundaries are also visible as appearance edges (though we note that many appearance edges arise solely due to surface markings or illumination effects). Neither motion nor appearance alone, however, is sufficient for the detection of occlusion boundaries. Accurate local motion estimates may be hard to obtain near occlusion boundaries, and appearance edges do not always correspond to occlusions. Thus we will combine multiple appearance cues, captured by state-of-the-art edge detectors, with local motion cues to show that *together* these distinct sources of information produce superior results to using either cue alone. In particular, our goal is to determine the subset of appearance edges that correspond to occlusion boundaries, thereby framing our problem as one of classification.

*This work was partially supported by a National Science Foundation Graduate Fellowship.

After explaining in Sections 2 and 3 the specifics of extracting our motion and appearance cues and their classification, we will describe our experiments in Section 4, demonstrating improved occlusion boundary detection when combining these cues. These experiments provide quantitative as well as anecdotal results on a novel dataset labeled for this task.

2 Local Occlusion Boundary Features

Edge detectors generally assign a "strength" to each pixel, which captures the degree to which an edge exists there, based on the contribution of various perceptual cues. At occlusion boundaries, there is often an additional cue in the form of inconsistent image motion. This motion may be caused by camera movement, which induces parallax at depth discontinuities, or it may be a result of dynamic objects in the scene. Our approach handles either situation equivalently and is thus more general than motion *detection* work that relies on a static camera for background subtraction, *e.g.* [14, 19]. In the following sections, we will describe our methods for extracting each of these features, which will then be used as cues for an occlusion boundary classifier described in Section 3.

2.1 Oriented Edge Detection

While classical edge detectors based on filtering are popular, most notably the Canny detector, they rely on rather simple models of image intensity at edges. Even moving beyond simple step edges to more complex edge types [13], linear filtering approaches still perform poorly on edges which exist between cluttered or textured regions. This is a serious concern for our work since we hope to extract motion in the vicinity of detected edges (as described in the Section 2.2 below). Motion is only observable when there is sufficient intensity gradient due to texture or clutter, so we need an edge detection approach which works well in such cases.

Thus we seek a detector capable of combining multiple cues which does not rely on overly simplistic edge models. An increasingly popular approach to achieve these goals computes edge strength using statistical comparisons of non-parametric distributions of cues on either side of a sample image patch at various orientations [8, 10, 11, 15, 20]. These detectors produce good results even on edges in texture and clutter and are therefore more appropriate for our task. Furthermore, they were extended to the spatio-temporal domain in [17], yielding a detector also capable of estimating an edge's normal *speed*. Though potentially useful for future work, here we focus instead on integrating *multiple appearance cues*, whereas [16, 17] only use intensity information.

Thus, we have chosen to use the popular Berkeley "*Pb*" detector for our experiments [10], which already incorporates three appearance cues (brightness, color, and texture) and offers a publicly available implementation. As an added benefit, the *Pb* detector's default parameters were learned on a large set of human-segmented data [9], allowing us to avoid tedious parameter tuning. At each location in the image, we interpolate better estimates for both orientation (θ) and edge strength (e) by fitting parabolas around the peak *Pb* response over the set of sampled orientations. Then we suppress those responses which are not local maxima along the edges' normal directions [13]. All edges which survive this suppression are kept for the classification step, *i.e.* we ignore edge strength at this stage (effectively thresholding at zero) to avoid prematurely ruling out edges simply because of low strength before also considering motion cues.

In Figure 1, we provide an example of edges detected using a traditional linear filtering approach (b), which is based on response to a quadrature pair of oriented filters [1, 5, 13],

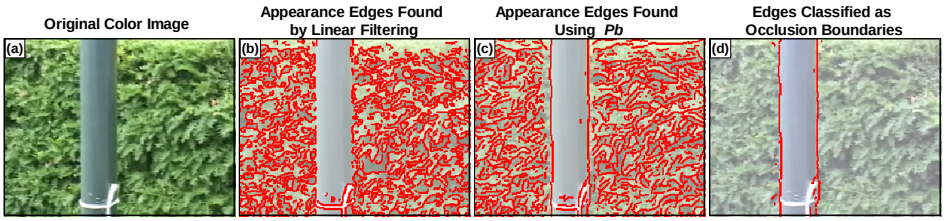


Figure 1: For an input image (a), we compare (b) classical edge detection using a quadrature pair of filters to (c) the Berkeley *Pb* detector (all non-zero responses after non-local maxima suppression are shown). Only the *Pb* detector fires consistently on the edges which lie on occlusion boundaries of the pole, giving subsequent classification a chance of succeeding. The goal of this work, then, is to utilize appearance and motion cues in order to classify which of those edge detections are also occlusion boundaries, as shown in (d).

as compared to the output of the *Pb* detector (c). Each shows all non-zero responses after non-local maxima suppression. Note how the *Pb* detector finds more consistent edges at the occlusion boundaries on the sides of the pole despite the background clutter. At this stage, we are most interested in providing all *potential* occlusion boundaries to the subsequent classifier (*i.e.* we can tolerate false positives but not false negatives). Therefore *Pb* is much better suited to our classification task, an example of which is shown in (d).

2.2 Local Multi-Frame Motion Estimation

As with edge detection, the estimation of image motion, *i.e.* optical flow, is a classical problem in computer vision (see [4] for a recent tutorial). Here, we will consider several consecutive frames of video and compute a *multi-frame* motion estimate. As compared to using only two frames, we find that using multiple frames produces substantially more robust estimates that are more discriminative for our classification task.

Given a set of frames $\{I^{(n)}\}_{n=-N}^{N}$ our goal is to find the translational motion, with components u and v, which best matches a patch of pixels P in the central reference image, $I^{(0)}$, with its corresponding patch in each of the other images, $\{I^{(n)}\}_{n\neq 0}$:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \arg\min \sum_{n=-N}^{N} h(n) \sum_{(x,y)\in P} w(x,y) \underbrace{\left(I^{(n)}(x+nu, y+nv) - I^{(0)}(x,y)\right)^2}_{I_t(u,v,n)} \quad (1)$$

This implicitly assumes constant translation for the duration of the set of frames, which we find to be reasonable over brief time periods.

We employ Gaussian-shaped weighting functions, $w(x,y)$ and $h(n)$ (with associated bandwidths σ_h and σ_w), to decrease the contribution spatially and temporally of pixels distant from the center of the reference patch. We iteratively estimate u and v using a multi-frame, Lucas-Kanade style differential approach. This amounts to solving iteratively the following least squares problem for new translation estimates (at iteration $k+1$), given the previous ones (at iteration k), based on spatial derivatives of the reference patch, I_x and I_y, and temporal derivatives, I_t:

$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_y I_x & \sum I_y^2 \end{bmatrix} \begin{bmatrix} u_{k+1} \\ v_{k+1} \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t(u_k, v_k, n) \\ \sum I_y I_t(u_k, v_k, n) \end{bmatrix}, \quad (2)$$

where the sums are taken over all pixels within the patch, across all frames. (For clarity, we have omitted the weights, $w(x,y)$ and $h(n)$, in this formulation.) In practice, we initially consider only $I^{(0)}$ and its two immediate neighbors. We then gradually increase the temporal window, initializing with the previous translation estimate, until finally considering all frames from $-N$ to N. This prevents frames at extremes of the temporal window from pulling us to poor local minima of (1)[1].

Aggregation of patches of data near occlusion boundaries is problematic and addressing this problem specifically for optical flow estimation is the subject of extensive research, including multiple motion estimation, robust estimators, line processes, and parametric models [2, 4]. Recently, impressive results computing dense flow fields in spite of significant occlusion boundaries by using a variational approach and bilateral filtering were demonstrated in [21].

For our purposes, since we are interested only in motion estimates near edges (rather than a dense flow field), we will choose patches of data P_L and P_R on either side of each detected edge pixel, as shown in Figure 2. In addition, because we have an estimate of each edge pixel's orientation, θ, we can align those windows to the edge in order to prevent the collection of information across a potential occlusion boundary. This technique is related to adaptive/multiple-window techniques, *e.g.* in stereo vision [6, 7], and was also recently used in occlusion reasoning [16]. (Spatio-temporal alignment to moving edges is also performed in [16], which could be used to augment our approach as well.) Computing the necessary derivatives within each window (via standard finite differencing), we can then estimate the motions ($\mathbf{u}_L = [\ u_L \quad v_L\ ]^T$ and $\mathbf{u}_R = [\ u_R \quad v_R\ ]^T$) of the patches on either side of each edge using the least squares approach outlined above. We then compute the difference in motion between the left and right patches, $\mathbf{u}_d = \mathbf{u}_L - \mathbf{u}_R$. Finally, we use the Euclidean norm of the $\mathbf{u}_d$ vector to capture the relative motion between the surfaces on either side of a potential occlusion boundary. This metric serves as the second feature, or cue, used by the classifier described in the next section.

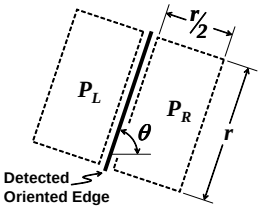


Figure 2: Patches for motion estimation aligned to an oriented edge.

In our experiments, this Euclidean metric proved to be just as useful as a Mahalanobis distance. This is likely due to the difficulty in obtaining good estimates of the necessary covariance information on the motion components (*e.g.* by using the Hessian in (2), which is not sufficient), without resorting to expensive sampling techniques [2]. More advanced motion estimation methods and distance metrics are possible avenues of continued research. For example, it may be useful to use an affine motion model or to consider separately the estimated components of motion normal and tangential to the edge's orientation.

3 Classification

Our goal is to label edges as occlusion boundaries or not. We do so by using the posterior probability of the existence of an occlusion boundary given our features, $\Pr(B|f)$, where f may represent the motion difference d, the edge strength e, or both $\{d,e\}$. Given the substantial, scene-dependent variation in the fraction of appearance edges that are also occlusion boundaries, we assume a uniform prior on $\Pr(B)$ and use Bayes' Rule to estimate

[1] This is equivalent to gradually increasing the bandwidth of $h(n)$.

this posterior (note that estimating a prior from the training data was not helpful):

$$\Pr(B|f) = \frac{p(f|B)}{p(f|B) + p(f|\neg B)}. \quad (3)$$

Given training data, we can sample our edge strength and motion difference freatures to estimate the necessary data likelihoods, $p(f|B)$ and $p(f|\neg B)$, as described in the next section. Thresholding this ratio yields the classifier used for our experiments. In the future, it may be possible to achieve better performance by learning adaptive priors for a given image sequence.

4 Experiments

We first need a dataset with labeled occlusion boundaries in order to learn the likelihoods for the classifier. Such a dataset currently does not exist[2]. Thus we have constructed a new dataset for this task, which will also be made available online for other researchers. It contains 30 short image sequences, approximately 8-20 frames in length with the ground truth occlusion boundaries labeled in the reference (*i.e.* middle) frame of each sequence. Some example scenes from this dataset are depicted in Figure 3 with their ground truth occlusion/object boundary labels overlaid. The dataset is quite challenging, with a variety of indoor and outdoor scene types, significant noise and compression artifacts, unconstrained handheld camera motions, and some moving objects. We plan to augment this dataset with further examples in the future.

Figure 3: Ground truth occlusion boundaries labeled for 12 of the 30 scenes from our dataset. Each example is the reference (middle) frame of a short sequence, usually 8-20 frames. The images have been lightened for clarity. The scene in Figure 1 is also in the dataset.

For our experiments, we first extract our edge strength feature by applying the Berkeley *Pb* code to the reference frame of each sequence, using all default parameters (*i.e.* those learned from the BSDS training data). Next we align each frame of the sequence to the reference frame using a global translational motion estimate, as suggested in [16]. This stabilization step removes gross camera motions, allowing us to focus on the (potentially small) *relative* patch motions which are most important for our task. In addition, the stabilized sequence better adheres to our constant velocity assumption. Then, as described in Section 2.2, we align small patches ($r = 12$ pixels) on either side of each edge according to the edges' detected orientations (see Figure 2). Using (1) and (2), we estimate the translational motion of each patch separately and compute the Euclidean distance between the two estimates. We use a temporal window radius of $N = 3$ frames and weighting function

[2]The popular Berkeley Segmentation Data Set (BSDS) [9] does not provide image *sequences* necessary for estimating motion, nor do the human-labeled edges necessarily correspond strictly to occlusion boundaries.

bandwidths of $\sigma_h = N$ and $\sigma_w = r$. As shown by the distribution in Figure 4, most relative motions $\mathbf{u}_d$ are quite small, with a mean of 0.14 pixels/frame. This supports our claim that the motion cue available for our task is quite subtle.

4.1 Training

We randomly select half of our dataset to use for training. We first determine the correct label for all detected edge pixels in an image by matching them to occlusion boundary pixels from the ground truth data. Because of localization inaccuracies (in labeling and detection), we use an approach similar in spirit to the one outlined in Appendix B of [10], which seeks to find a one-to-one correspondence between detected edge pixels and nearby hand-labeled boundary pixels. A given training set consists of 15 scenes, yielding a total of approximately $80,000$ individual examples of edge pixels for training. Unfortunately, these examples are taken from contiguous edges and therefore the patches used in generating their appearance and motion cues overlap significantly. Thus they are highly dependent samples, making it inappropriate to use them all for training.

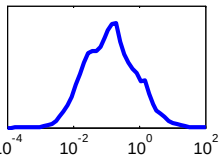


Figure 4: Empirical distribution of relative motions $\mathbf{u}_d$.

To alleviate this problem somewhat, we consider only a random subset of the edges available in the training set. This subset is selected such that no two samples which come from the same image could have utilized overlapping patches of data in estimating motion or computing *Pb*. Thus, for these experiments, we sample edges that are at least $r = 12$ pixels apart. The resulting subset contains approximately 6000 examples, which we use for the training described below. (For testing in Section 4.2, we classify *all* edges detected in a given image.)

Using the edge strength and motion features for all edge pixels corresponding to ground truth occlusion boundaries, we construct kernel density estimates of each cue likelihood independently, $p(e|B)$ and $p(d|B)$, as well as their joint likelihood, $p(e,d|B)$. Similarly, we use any detected edges that are *not* occlusion boundaries as negative examples to learn $p(e|\neg B)$, $p(d|\neg B)$, and $p(e,d|\neg B)$. We use a Gaussian kernel with $\sigma = 1$ bin, and $\pm 3\sigma$ support. For each cue, we use 50 bins (and thus the joint likelihood estimate contains 50×50 bins). In our experience, using a kernel does offer improved results, despite the fairly coarse binning, particularly in terms of generalization from training to test data. To emphasize the importance of distinguishing the very small motion differences (Figure 4), the bins used for estimating the likelihood of the motion-difference cue are logarithmically spaced between 10^{-3} and 10^{2} (where very large motion is indicative of noise or lack of texture). The bins for edge strength are linearly spaced between 0 and 1.

The resulting independent cue likelihoods are shown in Figure 5. As evidenced by the separation of the distributions for each class, these cues do contain some distinct information for our classification task. The distributions also make intuitive sense: higher edge strength and larger motion differences more commonly correspond to occlusion boundaries. It is worth noting that the motion difference cue is fairly weak (*i.e.* the distributions overlap significantly). While improved motion estimation techniques may help, this further supports our claim that the use of optical flow alone for finding occlusion boundaries, as is common practice in segmentation schemes based on motion, could produce poor results on natural scenes which lack texture at many true occlusion boundaries.

The estimated joint likelihoods are shown in Figure 6. We have estimated the full two-dimensional joint distributions $p(e,d|B)$ and $p(e,d|\neg B)$ as well as approximate joint distributions $p(e|B)p(d|B)$ and $p(e|\neg B)p(d|\neg B)$, which assume our two cues are inde-

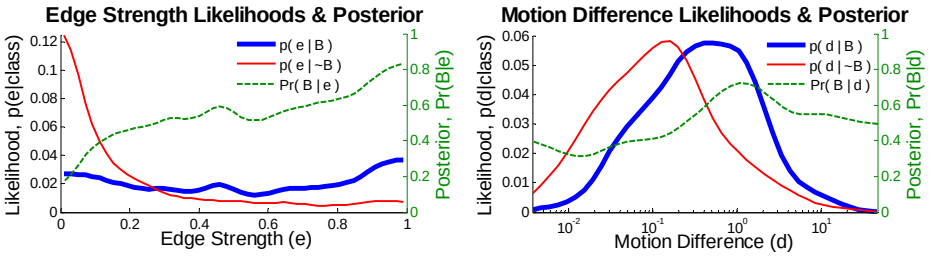


Figure 5: Independent distributions and ratio scores for our two cues.

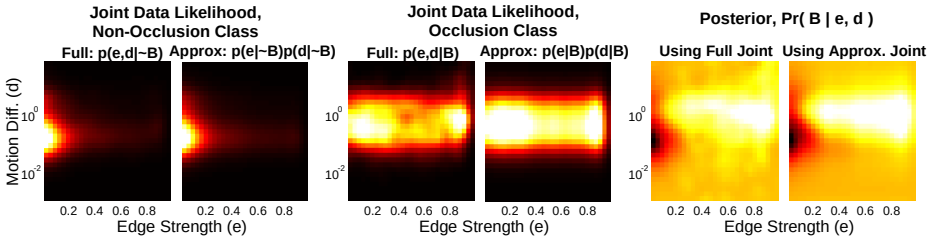


Figure 6: Learned joint likelihood distributions and ratio scores for our two cues. Left and right of each pair shows the result using the full and approximated joint, respectively.

pendent. Given the visually similar estimates, it would appear safe to make such an independence assumption and approximate the joint in this manner. We will test our classifier with both versions below.

Next we compute the posterior probability according to (3). For the separate cues, the result is overlaid on the likelihoods in Figure 5. For the combined cues, the posterior estimates are found in the rightmost pair of Figure 6. Rather than fitting an arbitrary model to the posterior, we have chosen to use the estimates as non-parametric lookup tables.

Finally, we evaluate the learned classifier on the training data itself. After estimating $\Pr(B|f)$ at each edge pixel, we generate Precision vs. Recall curves by varying the threshold on that posterior estimate and counting the number that were correctly labeled. As seen in the left plot of Figure 7, each cue separately provides some information, but the two together perform better, with the full joint providing the best result. The precision levels of these curves also capture a notion of the difficulty of our task and dataset.

We can repeat the entire training process with a different randomly-selected set of sequences for training. Doing so allows us to compute the error bars on the precision recall curves show in Figure 7. These error bars represent plus/minus one standard deviation ($\hat{\sigma}$) for $n = 50$ trials. Thus they indicate the typical distribution of the curves for various divisions of the data. The confidence intervals based on standard errors ($\hat{\sigma}/\sqrt{n}$) are very tight and visually imperceptible from the mean (and thus are not shown). This indicates a statistically significant difference between the mean curves in the plots.

4.2 Testing

For testing, we use the remainder of the dataset, extracting motion and edge strength cues as before. This includes the other half of the scenes, again with approximately $80,000$ examples to be classified. We classify each edge pixel by thresholding the estimated posterior. We can vary this threshold to produce the Precision vs. Recall curves shown in the right plot of Figure 7. Here we see confirmation that the learned classifier can generalize

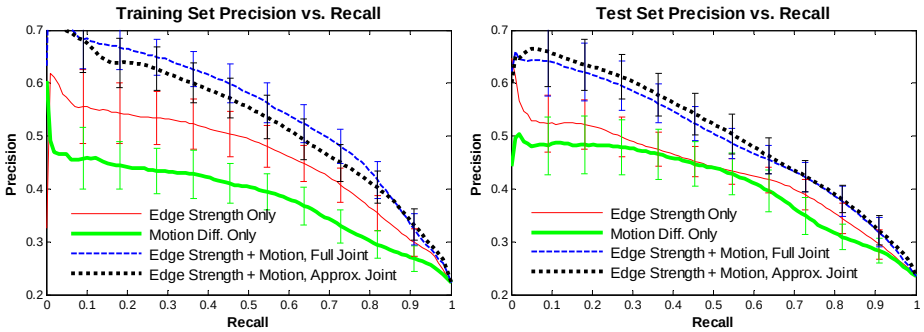


Figure 7: Precision vs. Recall curves for the training and test sets (left and right, respectively), using various combinations of cues. Error bars indicate plus/minus one standard deviation of the curves for 50 randomly selected divisions of the dataset.

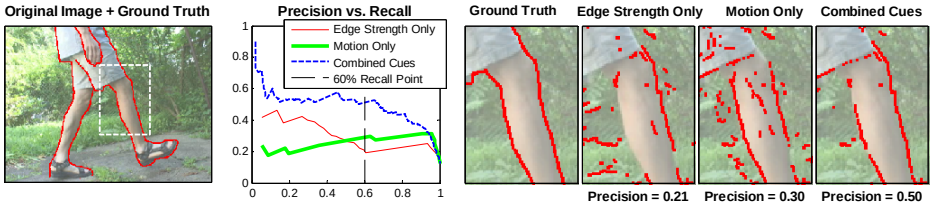


Figure 8: Example classification result at a chosen operating point of 60% recall. Combining appearance and motion cues produces superior precision than either cue alone. Note in the combined result the increased detection on the left of the leg as compared to using edge strength alone, and the decreased spurious detections as compared to using motion alone.

to novel scenes. We see similar performance between the full and approximated joint distributions, with marginal improvement using the approximation. This may indicate that the full joint estimate is slightly overfitting the training data. And once again, by repeating the experiment with different test sets, we can generate the displayed error bars.

Aggregated results as provided in Figure 7 give a general sense of performance, but here we also provide a few anecdotal examples from the dataset to exhibit more concretely the information sometimes hidden in such cumulative comparisons. Figure 8 shows a scene with ground truth overlaid. To illustrate the improvement when using both cues together, we have selected the threshold for each classifier that results in 60% recall, as indicated on the Precision vs. Recall plot. For the indicated window of the original scene, the right four boxes compare the ground truth labeling and the classification results using the cues individually and together. As shown, the best result (with significantly higher precision) is achieved using both cues. For example, combined cues yield improved detection with fewer false positives on the left side of the leg as compared to the result using individual cues alone. Similarly, the examples in Figure 9 demonstrate classification improvement using combined cues.

5 Discussion & Conclusion

Because the performance of any local edge detector is limited, some edges will always be missed. By restricting ourselves to the classification of only the appearance edges

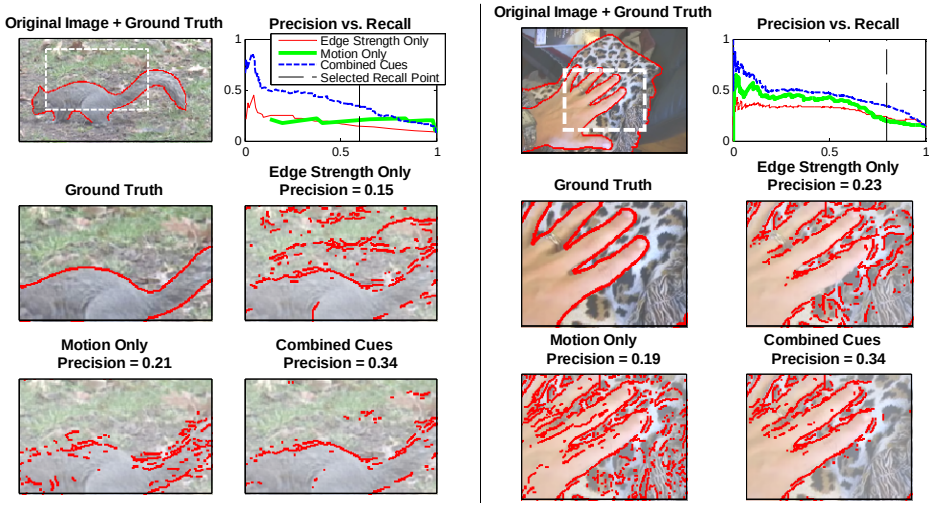


Figure 9: Two additional example classification results. Combining appearance and motion cues produces superior precision at the selected recall operating point than using either cue alone. Note the decreased false positives when using the combined cues.

which *are* detected, we therefore inherit those limitations. As detectors improve (*e.g.* in detecting very weak edges), so too will our approach. For our dataset, however, the edge detector fires with non-zero strength on 83.5% of the ground truth boundaries, indicating that our technique is viable in practice. A complementary approach may include finding motion boundaries *first* and subsequently incorporating appearance reasoning. This may allow the detection of occlusion boundaries visible *only* due to motion, but these cases are relatively rare and such an approach could come at high computational cost.

Local estimates of any kind, including the *Pb* detector and our motion difference feature, are inherently noisy and ambiguous. They are most useful when incorporated into more global reasoning, *e.g.* using a graphical model. Rather than blindly using local estimates for mid- and high-level tasks, however, we believe it is important, if not crucial, to evaluate the utility of these low-level cues themselves (separately and in combination). Having verified here the benefit of using motion, we are currently developing methods of globally reasoning about object/occlusion boundaries and object segmentation which build on the combined local cues described in this work.

Our goal of detecting occlusion boundaries could potentially benefit many computer vision methods, which often rely on spatial aggregation. In this work, we have presented experiments demonstrating anecdotal and quantitative results for two local, low-level feature types useful for future research into globally reasoning about occlusion boundaries. While further investigation into augmenting and strengthening each of our chosen features is warranted, particularly in the better estimation and comparison of local motion, we have demonstrated that considerable improvement in classifying occlusion boundaries is possible when combining these two distinct, individually weaker cues. We have also provided a novel, labeled dataset as an additional resource for future research on occlusion boundary detection.

Acknowledgements

The authors thank David Fleet for his many helpful comments and suggestions.

References

[1] Edward H. Adelson and James R. Bergen. Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America A*, 2(2):284–299, February 1985.

[2] Michael J. Black and David J. Fleet. Probabilistic detection and tracking of motion discontinuities. *IJCV*, 38(3):231–245, 2000.

[3] Doron Feldman and DaphnaWeinshall. Motion segmentation using an occlusion detector. In *Worksop on Dynamical Vision at ECCV*, 2006.

[4] David J. Fleet and Yair Weiss. Optical flow estimation. In N. Paragios, Y. Chen, and O. Faugeras, editors, *Mathematical models for Computer Vision: The Handbook*. Springer, 2005.

[5] W. T. Freeman and E. H. Adelson. The design and user of steerable filters. *PAMI*, 13(9):891–906, September 1991.

[6] Heiko Hirschmüller, Peter R. Innocent, and Jon Garibaldi. Real-time correlation-based stereo vision with reduced border errors. *IJCV*, 47(1-3):229–246, April-June 2002.

[7] Takeo Kanade and Masatoshi Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. *PAMI*, 16(9):920–932, September 1994.

[8] Scott Konishi, Alan L. Yuille, James M. Coughlan, and Song Chun Zhu. Statistical edge detection: Learning and evaluating edge cues. *PAMI*, 25(1):57–74, January 2003.

[9] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, volume 2, pages 416–423, July 2001.

[10] David R. Martin, Charless C. Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *PAMI*, 26(5):530–549, May 2004.

[11] Bruce A. Maxwell and Stephanie J. Brubaker. Texture edge detection using the compass operator. In *BMVC*, volume II, pages 549–558, September 2003.

[12] Abhijit S. Ogale, Cornelia Fermüller, and Yiannis Aloimonos. Motion segmentation using occlusions. *PAMI*, 27(6):988–992, June 2005.

[13] Pietro Perona and Jitendra Malik. Detecting and localizing edges composed of steps, peaks, and roofs. In *ICCV*, pages 52–57, 1990.

[14] Michael G. Ross and Leslie Pack Kaelbling. Learning static object segmentation from motion segmentation. In *AAAI*, 2005.

[15] Mark Ruzon and Carlo Tomasi. Color edge detection with the compass operator. In *CVPR*, pages 160–166, June 1999.

[16] Andrew N. Stein and Martial Hebert. Local detection of occlusion boundaries in video. In *BMVC*, 2006.

[17] Andrew N. Stein and Martial Hebert. Using spatio-temporal patches for simultaneous estimation of edge strength, orientation, and motion. In *Beyond Patches Workshop at CVPR*, 2006.

[18] Regis Vaillant and Olivier D. Faugeras. Using extremal boundaries for 3-D object modeling. *PAMI*, 14(2):157–173, 1992.

[19] Thomas Veit, Frédéric Cao, and Patrick Bouthemy. An a contrario decision framework for region-based motion detection. *IJCV*, 68(2):163–178, June 2006.

[20] Lior Wolf, Xiaolei Huang, Ian Martin, and Dimitris Metaxas. Patch-based texture edges and segmentation. In *ECCV*, 2006.

[21] Jiangjian Xiao, Hui Cheng, Harpreet Sawhney, Cen Rao, and Michael Isnardi. Bilateral filtering-based optical flow estimation with occlusion detection. In *ECCV*, volume I, pages 211–224, 2006.

Motion Segmentation Using Inference in Dynamic Bayesian Networks

Marc Toussaint
TU Berlin
Franklinstr. 28/29
10587 Berlin, Germany
mtoussai@cs.tu-berlin.de

Volker Willert, Julian Eggert, Edgar Körner
Honda Research Institute Europe GmbH
Carl-Legien-Str. 30
D-63073 Offenbach/Main, Germany
volker.willert@honda-ri.de

Abstract

Existing formulations for optical flow estimation and image segmentation have used Bayesian Networks and Markov Random Field (MRF) priors to impose smoothness of segmentation. These approaches typically focus on estimation in a single time slice based on two consecutive images. We develop a motion segmentation framework for a continuous stream of images using inference in a corresponding Dynamic Bayesian Network (DBN) formulation. It realises a spatio-temporal integration of optical flow and segmentation information using a transition prior that incorporates spatial and temporal coherence constraints on the flow field and segmentation evolution. The main contribution is the embedding of these particular assumptions into a DBN formulation and the derivation of a computationally efficient two-filter inference method based on factored belief propagation (BP) that allows for on- and offline parameter optimisation. The spatio-temporal coupling implemented in the transition priors ensures smooth flow field and segmentation estimates without using MRFs. The algorithm is tested on synthetic and real image sequences.

1 Introduction

Optical flow estimation is a fundamental problem in image processing. The analysis of movement in the image allows one to infer the motion of objects in the environment as well as the self-motion relative to the environment. As it is generally the case in information processing problems, the quality of the estimation can be greatly enhanced when information from different sources is integrated. In image sequences, an important source of information is prior knowledge about the structure of optical scenes. One may assume that images are composed of segments which refer to different physical objects. Each object induces a coherent flow field in its segment reflecting its 3D motion. An elegant way to integrate structural

segmentation and flow field estimation is to formulate a generative probabilistic model in terms of Bayesian Networks [13, 3, 11, 10]. These existing approaches focus on a single time slice flow field coupled to two consecutive images and usually implement smoothness of segmentation using Markov Random Field (MRF) priors [4, 3, 5, 9]. This can be applied on continuous image sequences by applying the technique on each time slice. However, this would neglect an additional source of information: the temporal (Markovian) coupling of the flow field and segmentation evolution. Further, when the application requires online motion and segmentation filtering, the computational cost of MRF inference in each time step may be too high [11]. Other approaches take the temporal coupling into account but do not consider the segmentation problem [1, 2, 12].

Our focus is on the temporal coupling of the flow field and segmentation in image sequences, and we avoid using intra-time slice MRF priors but instead achieve smoothness with a proper spatio-temporal coupling. A crucial aspect is formulating appropriate transition probabilities for the flow field and segmentation evolution. We formulate transition probabilities that allow us to propagate information across time slices but also imply a spatial smoothness of the flow field estimation and segmentation. Hence, instead of iterating BP (during MRF inference) within one time slice to impose spatial smoothness, we "unroll" this constraint and encode it in the spatio-temporal coupling such that BP along a temporal sequence yields spatial smoothness after a few time steps. This approach is particularly well-suited for online applications since the computational cost at each time step is limited to a small and fixed number of message passings.

After we have formulated the DBN model and the corresponding transition probabilities in section 2 we describe in section 3 our inference and parameter learning algorithm. During inference we make several factorisation assumptions and use a version of the Factored Frontier Algorithm [6]. For parameter training using a batch or online EM-algorithm we use either a two-filter or a forward filter approach to compute the flow field and segmentation posteriors. Based on the computed posteriors we update parameters in an M-step. Section 4 discusses experimental results and section 5 concludes the paper.

2 Dynamic Bayesian Network Model

We start by specifying a complete data likelihood of a sequence $I^{0:T}$ of $T+1$ images. We do this by assuming the generative model for such an image sequence as given by the DBN in Fig. 1A. Here, I^t is the grey value image at time slice t with entries I^t_x that are the grey values at all pixel locations $x \in X$ of the image. Similarly, V^t is a flow field at time slice t defined over the image range with entries $v^t_x \in W$ at each pixel location x of the image. Throughout this paper we consider discrete flow fields ($W = \mathbb{Z}^2$). Further, we assume that each image is composed by a finite number of segments. Each segment has a shape and the discrete labelling $s^t_x \in \{1, .., K\}$ of each pixel specifies which image pixels stem from which of K possible segments. Since every segment has a typical vector field describing the optical flow of its appearance, the segment labelling variable S^t is coupled to the flow field variable V^t.

To define the model precisely we need to specify (i) the observation likelihood $P(I^{t+1} \,|\, V^t, I^t)$ of a pair of images I^{t+1} and I^t, (ii) the transition probability

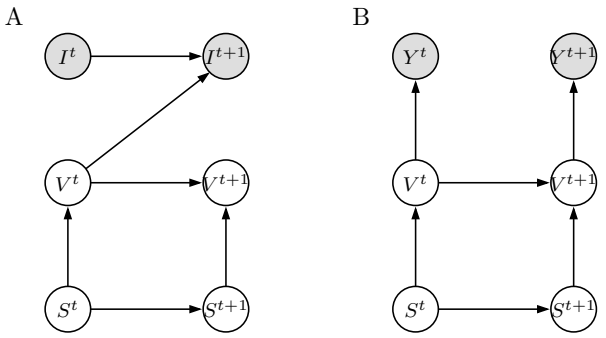


Figure 1: Dynamic Bayesian Network for motion estimation.

$P(V^{t+1} \,|\, S^{t+1}, V^t)$ of the flow field depending on the segmentation, and (iii) the transition probability $P(S^{t+1} \,|\, S^t)$ of the segmentation.

To simplify the notation we can introduce an alternative observation variable $Y^t = (I^{t+1}, I^t)$ that subsumes a pair of consecutive images. Since images are observed, the likelihood $P(I^t)$ in the term $P(I^{t+1} \,|\, V^t, I^t)\ P(I^t) = P(I^{t+1}, I^t \,|\, V^t)$ is only a constant factor we can neglect. This leads to the DBN shown in Fig. 1B with observation likelihoods $P(I^{t+1} \,|\, V^t, I^t) \propto P(I^{t+1}, I^t \,|\, V^t) = P(Y^t \,|\, V^t)$. For all transition probabilities we assume that they factorise over the image as follows,

$$P(Y^t \,|\, V^t) = \prod_x \ell(Y^t \,|\, v_x^t)\ , \tag{1}$$

$$P(V^{t+1} \,|\, S^{t+1}, V^t) = \prod_x P(v_x^{t+1} \,|\, S^{t+1}, V^t)\ , \tag{2}$$

$$P(S^{t+1} \,|\, S^t) = \prod_x P(s_x^{t+1} \,|\, S^t)\ . \tag{3}$$

2.1 Observation likelihood

We define the observation likelihood $P(Y^t \,|\, V^t)$ by assuming that the likelihood $\ell(Y^t \,|\, v_x^t)$ of a local velocity v_x^t should be related to finding the same or similar image patch $\mathbf{I}_x^{t+1}$ centred around x at time $t+1$ that was present at time t but centred around $x - v_x^t \Delta t$. In the following, we neglect dimensions and set $\Delta t = 1$. Let $\mathcal{S}(x, \mu, \Sigma, \nu)$ be the Student's t-distribution and $\mathcal{N}(x, \mu, \Sigma) = \lim_{\nu \to \infty} \mathcal{S}(x, \mu, \Sigma, \nu)$ be the normal distribution of a variable x with mean μ, covariance matrix Σ and the degrees of freedom ν. We define

$$\ell(Y^t \,|\, v_x^t) = \mathcal{S}(\mathbf{I}_x^{t+1}, \mathbf{I}_{x - v_x^t}^t, \frac{\sigma_I^2}{\mathcal{N}(x, x', \varrho_I)}, \nu_I)\ . \tag{4}$$

In this notation, the image patches can be regarded vectors and the covariance matrix is a diagonal with entries $\sigma_I^2 / \mathcal{N}(x, x', \varrho_I)$ that depend on the position x' relative to the centre (i.e., a heteroskedastic variance). In effect, the term $\mathcal{N}(x, x', \varrho_I)$ implements a Gaussian weighting of locality centred around x for the patch $\mathbf{I}_x^{t+1}$ and around $x - v_x^t$ for the patch $\mathbf{I}_{x - v_x^t}^t$. The parameter ϱ_I defines the spatial range

of the image patches and σ_I the grey value variance. The univariate Student's t-distribution realises a robust behaviour against large grey-value differences within image patches, which means the euclidean distance between the two patches is treated as an outlier if it is too large.

2.2 Flow field transition probability

The definition of the flow field transition probability $P(V^{t+1} \,|\, S^{t+1}, V^t)$ includes two factors. First, we assume that the flow field transforms according to itself. Second, the segmentation imposes an additional factor on the transition probability acting similarly to a prior over V^{t+1} depending on the segmentation.

Let us first discuss the first factor: We assume that the origin of a local flow vector v_x^{t+1} at position x was a previous flow vector $v_{x'}^t$ at some corresponding position x',

$$v_x^{t+1} \sim \mathcal{S}(v_x^{t+1}, v_{x'}^t, \sigma_V, \nu_V) \ . \tag{5}$$

So, we assume robust spatio-temporal coherence because evaluations on first derivative optical flow statistics [7] and on prior distributions that allow to imitate human speed discrimination tasks [8] provide strong indication that they resemble heavy tailed distributions. Now, asking what the corresponding position x' in the previous image was, we assume that we can infer it from the flow field itself via

$$x' \sim \mathcal{N}(x', x - v_x^{t+1}, \varrho_V) \ . \tag{6}$$

Note that here we use v_x^{t+1} to retrieve the previous corresponding point. Combining both factors and integrating x' we would get (still neglecting the coupling to the segmentation)

$$v_x^{t+1} \,|\, V^t \sim \sum_{x'} \mathcal{N}(x', x - v_x^{t+1}, \varrho_V) \ \mathcal{S}(v_x^{t+1}, v_{x'}^t, \sigma_V, \nu_V) \ . \tag{7}$$

The parameter ϱ_V defines the spatial range of a flow-field patch, so we compare velocity vectors within flow-field patches at different times t and $t+1$.

We introduced new parameters ϱ_V and σ_V for the uncertainty in spatial identification between two images and the transition noise between V^t and V^{t+1}, respectively. The robustness against outliers is controlled by ν_V, with smaller/larger ν_V decreasing/increasing the influence of incoherently moving pixels within the observed spatial range ϱ_V.

So far we have not discussed the coupling of the segmentation to the flow field transition. Each segment corresponds to a typical flow pattern $q_s(V^t)$. On its own (i.e., within one time slice), a flow pattern q_s corresponds to a prior over the flow field,

$$q_s(V^t) = \prod_x \mathcal{N}(v_x, A_s x + t_s, \sigma_s) \ . \tag{8}$$

This is equivalent to assuming the world is approximately a set of planar objects such that each of their movements are completely described by an affine parameterisation A_s and t_s, with A_s being a 2×2 matrix describing any combination of rotation, divergence and shear, and t_s being a 2×1 translational vector.

The segmentation field S^t, which contains for every pixel a label $s_x^t \in \{1, .., K\}$, specifies the correspondence of each pixel to each flow pattern. This segmentation couples to the flow field transition probability as an additional factor. Combining this with (7) we finally define the flow field transition probability as

$$P(v_x^{t+1} \,|\, S^{t+1}, V^t) \propto q_{s_x^{t+1}}(v_x^{t+1}) \sum_{x'} \mathcal{N}(x', x - v_x^{t+1}, \varrho_V)\ \mathcal{S}(v_x^{t+1}, v_{x'}^t, \sigma_V, \nu_V)\ . \tag{9}$$

2.3 Segmentation transition probability

For the transition of the segmentation field itself we assume, following exactly the same reasoning as previously for $v_x^{t+1} \,|\, V^t$ in equation (7),

$$P(s_x^{t+1} \,|\, S^t) \propto \sum_{x'} \mathcal{N}(x', x - \bar{q}_{s,x}^{t+1}, \varrho_S)\ Q(s_x^{t+1}, s_{x'}^t, \nu_S)\ , \tag{10}$$

where $\bar{q}_{s,x}^{t+1}$ is the mean of $q_{s_x^{t+1}}$ at x, and $Q(s, s', \nu_S)$ adds uniform noise on a discrete random variable,

$$Q(s, s', \nu_S) = \begin{cases} 1 - \nu_S(K-1) & \text{for s'=s} \\ \nu_S & \text{otherwise} \end{cases}\ . \tag{11}$$

That is, we assume that the segmentation field transforms according to the flow field priors q_s as specified by the segmentation itself; when a pixel is labelled s we expect that it transforms according to the flow field $\bar{q}_s$.

3 Factored frontier inference

The finite set of flow patterns q_s is a basic mechanism to introduce global (w.r.t. the image range, not the time horizon) structure to the prior over flow fields, similar to a mixture of factored models. As a consequence, the exact belief over the flow field (e.g., during filtering) and the segmentation will not decouple. Dealing with a full joint distribution over the flow field V^t and the segmentation S^t is infeasible. Hence, we will use an approximate inference technique based on factored belief propagation, which can be regarded as a factored frontier algorithm [6]. The factored observation likelihoods and transition probabilities we introduced ensure that the forward propagated messages will remain factored as well. We will describe the algorithm here in the jargon of loopy BP. Below we briefly restate the factored frontier interpretation of the algorithm.

We start by assuming the belief over V^t and S^t at time t to be factored,

$$P(V^t, S^t \,|\, Y^{1:t}) = P(V^t \,|\, Y^{1:t})\ P(S^t \,|\, Y^{1:t}) =: \alpha(V^t)\ \alpha(S^t) = \prod_x \alpha(v_x^t)\ \alpha(s_x^t)\ . \tag{12}$$

We first propagate a message forward from S^t to S^{t+1}, resulting in

$$\widetilde{\alpha}(S^{t+1}) = \prod_x \widetilde{\alpha}(s_x^{t+1})\ , \quad \widetilde{\alpha}(s_x^{t+1}) \propto \mu_{s \to s'}(s_x^{t+1}) \tag{13}$$

$$\begin{aligned}\mu_{s\to s'}(s_x^{t+1}) &\propto \sum_{S^t\in\{1,..,K\}^X} P(s_x^{t+1}\,|\,S^t)\ \alpha(S^t)\\ &= \sum_{S^t}\sum_{x'} \mathcal{N}(x', x-\bar{q}_{s,x}^{t+1}, \varrho_S)\ Q(s_x^{t+1}, s_{x'}^t, \nu_S)\ \prod_z \alpha(s_z^t) \qquad (14)\\ &= \sum_{x'} \mathcal{N}(x', x-\bar{q}_{s,x}^{t+1}, \varrho_S)\ \sum_{s_{x'}^t} Q(s_x^{t+1}, s_{x'}^t, \nu_S)\ \alpha(s_{x'}^t) \underbrace{\sum_{S^t\setminus s_{x'}^t}\prod_{z\neq x'}\alpha(s_z^t)}_{=1}\ .\end{aligned}$$

Note that the summation $\sum_{S^t\in\{1,..,K\}^X}$ is summing over all possible segmentation fields (X is the pixel range), i.e. it represents $|X|$ summations $\sum_{s_1^t}\sum_{s_2^t}\sum_{s_3^t}\cdots$ over each local segmentation label. We separated these into a summation $\sum_{s_{x'}^t}$ over the label at x' and a summation $\sum_{S^t\setminus s_{x'}^t}$ over all other labels at $x\neq x'$. Hence we can use $\sum_{S^t\setminus s_{x'}^t}\prod_{z\neq x'}\alpha(s_z^t) = \prod_{z\neq x'}\sum_{s_z^t}\alpha(s_z^t) = 1$.

Next we propagate forward from V^t, S^{t+1}, and Y^{t+1} to V^{t+1} resulting in

$$\begin{aligned}\alpha(V^{t+1}) &= \prod_x \alpha(v_x^{t+1})\ , \quad \alpha(v_x^{t+1}) \propto \ell(Y^{t+1}\,|\,v_x^{t+1})\ \mu_{v\to v'}(v_x^{t+1})\ \mu_{s\to v}(v_x^{t+1})\ ,\\ \mu_{s\to v}(v_x^{t+1}) &\propto \sum_{s_x^{t+1}} q_{s_x^{t+1}}(v_x^{t+1})\ \widetilde{\alpha}(s_x^{t+1})\ , \qquad (15)\\ \mu_{v\to v'}(v_x^{t+1}) &\propto \sum_{V^t\in W^X}\sum_{x'} \mathcal{N}(x', x-v_x^{t+1}, \varrho_V)\ \mathcal{S}(v_x^{t+1}, v_{x'}^t, \sigma_V, \nu_V)\ \prod_z \alpha(v_z^t)\\ &= \sum_{x'} \mathcal{N}(x', x-v_x^{t+1}, \varrho_V)\ \sum_{v_{x'}^t} \mathcal{S}(v_x^{t+1}, v_{x'}^t, \sigma_V, \nu_V)\ \alpha(v_{x'}^t)\ , \qquad (16)\end{aligned}$$

analogous to (14). Finally we pass back a message from V^{t+1} to S^{t+1},

$$\begin{aligned}\alpha(S^{t+1}) &= \prod_x \alpha(s_x^{t+1})\ , \quad \alpha(s_x^{t+1}) \propto \widetilde{\alpha}(s_x^{t+1})\ \mu_{v\to s}(s_x^t) \qquad (17)\\ \mu_{v\to s}(s_x^t) &\propto \sum_{v_x^{t+1}} q_{s_x^{t+1}}(v_x^{t+1})\ \frac{\alpha(v_x^{t+1})}{\mu_{s\to v}(v_x^{t+1})}\\ &\propto \sum_{v_x^{t+1}} q_{s_x^{t+1}}(v_x^{t+1})\ \ell(Y^{t+1}\,|\,v_x^{t+1})\ \mu_{v\to v'}(v_x^{t+1})\ . \qquad (18)\end{aligned}$$

This inference procedure can be interpreted as a Factor Frontier Algorithm (FFA) as follows. We start with a factored frontier belief (12) over V^t and S^t. The FFA first adds the node S^{t+1} to formulate the joint $P(V^t, S^t, S^{t+1})$. For this joint the marginals over V^t and S^{t+1} are computed and these define the new factored frontier. In this step it turns out that the marginal over V^t remains unchanged whereas the marginal over S^{t+1} is our $\widetilde{\alpha}(S^{t+1})$ we computed in equation (13). To the new frontier, the node V^{t+1} is added to formulate the joint over $P(V^t, S^{t+1}, V^{t+1})$ (including the new observation likelihood factor $P(Y^{t+1}\,|\,V^{t+1})$). For this joint the marginals over S^{t+1} and V^{t+1} are computed to yield the new factored frontier. It turns out that these marginals are $\alpha(S^{t+1})$ and $\alpha(V^{t+1})$ as we computed them in (17) and (15).

3.1 Two-filter inference

If we have access to a batch of data (or a recent window of data) we can compute smoothed posteriors as a basis for an EM-algorithm and train the free parameters. In our two-filter approach we derive the backward filter as a mirrored version of the forward filter, using

$$P(v_x^t \,|\, S^t, V^{t+1}) \propto q_{s_x^t}(v_x^t) \; \sum_{x'} \mathcal{N}(x', x + v_x^t, \varrho_V) \; \mathcal{S}(v_x^t, v_{x'}^{t+1}, \sigma_V, \nu_V) \tag{19}$$

$$P(s_x^t \,|\, S^{t+1}) \propto \sum_{x'} \mathcal{N}(x', x + \bar{q}_{s,x}^t, \varrho_S) \; Q(s_x^t, s_{x'}^{t+1}, \nu_S) \; . \tag{20}$$

instead of (9) and (10). These equations are motivated in exactly the same way as we motivated (7): e.g., we assume that the $v_x^t \sim \mathcal{S}(v_x^t, v_{x'}^{t+1}, \sigma_V, \nu_V)$ for a corresponding position x' in the subsequent image, and that $x' \sim \mathcal{N}(x', x - v_x^t, \varrho_V)$ is itself defined by v_x^t. However, note that using this symmetry of argumentation is actually an approximation to our model because applying Bayes rule on (9) or (10) would lead to a different, non-factored $P(V^t \,|\, S^t, V^{t+1})$. The backward filter equations are exact mirrors of the forward equations. To derive the smoothed posterior we need to combine the forward and backward filters. In the two-filter approach this reads

$$\begin{aligned} \gamma(v_x^t) \;\; &:= \;\; P(v_x^t \,|\, Y^{1:T}) = \frac{P(Y^{t+1:T} \,|\, v_x^t) \; P(v_x^t \,|\, Y^{1:t})}{P(Y^{1:T})} \\ &= \;\; \frac{P(v_x^t \,|\, Y^{t+1:T}) P(Y^{t+1:T}) P(v_x^t \,|\, Y^{1:t})}{P(v_x^t) P(Y^{1:T})} \propto \alpha(v_x^t) \; \beta(v_x^t) \; \frac{1}{P(v_x^t)} \end{aligned} \tag{21}$$

with $P(Y^{t+1:T})$ and $P(Y^{1:T})$ being constant. If both the forward and backward filters are initialised with $\alpha(v_x^0) = \beta(v_x^T) = P(v_x)$ we can identify the normalisation $P(v_x^t)$ with the prior $P(v_x)$. The same holds for the smoothed segmentation estimate $\gamma(s_x^t)$.

3.2 Parameter adaptation

This paper focuses on adapting the parameters that define each segment. These are the 6 parameters A_s and t_s and the variance σ_s associated with each of the K segment models q_s. Following the EM-algorithm we use inference on an image sequence to derive posteriors $\gamma(v_x^t)$ and $\gamma(s_x^t)$ over the latent variables $V^{1:T}$ and $S^{1:T}$. The exact M-step would then compute parameters that maximise the expected data log-likelihood, expectations taken by integrating over all latent variables. However, given the high-dimensionality of our latent variables a full integration is computationally too expensive. Thus, we use approximate M-steps based on the MAP estimates of the latent variables.

More precisely, the estimation of A_s and t_s are based on the MAP flow field estimates $\hat{v}_x^t = MAP(\gamma(v_x^t))$. Since q_s is assumed to be Gaussian, parameter estimates can be found by weighted linear regression of the MAP flow field. Further, let $w_{s,x}^t = 1$ if the pixel x at time t corresponds most likely to segment s, i.e. iff $\hat{s}_x^t = MAP(\gamma(s_x^t)) = s$, and zero otherwise. Then we update the variance σ_s as

$$\sigma_s = \frac{\sum_{x,t} \gamma(\hat{v}_x^t) \; w_{s,x}^t \; (\hat{v}_x^t - \bar{q}_{s,x}^t)^2}{\sum_{x,t} \gamma(\hat{v}_x^t) \; w_{s,x}^t} \; . \tag{22}$$

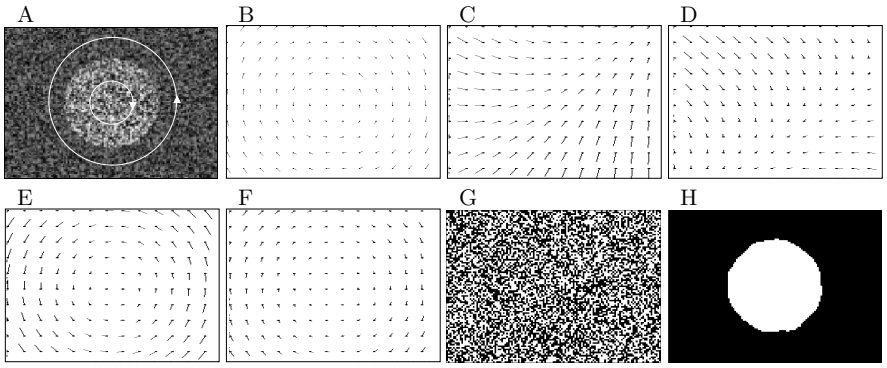


Figure 2: Rotating disc example.

4 Examples

We first test the algorithm on a scene composed of a rotating disc on a rotating background. Fig. 2A shows an image with a circular random pattern in the centre of the image rotating with constant angular velocity of $\omega = -\pi/30\Delta t$ and a random background pattern rotating with $\omega = \pi/60\Delta t$. The sequence contains 5 images. Fig. 2B displays the MAP flow field after 5 EM iterations using two-filter inference on the sequence batch. The pair of Figures 2C&D shows the mean of the random initialisation of q_s for both segments, while the pair of Figures 2E&F displays q_s for both segments after the 5 EM iterations. The two flow patterns q_s nicely specialised to model either the left or right rotating fields (the right rotation with double velocity). Finally, 2G displays the initial random segmentation of the image while 2H shows the final MAP segmentation after 5 EM iterations, which is close to the best possible pixel-based accuracy.

Next we tested the algorithm on the Flower Garden sequence. Fig. 3A displays the 15^{th} image of the original sequence. In a first experiment we allowed for 4 different segments ($K = 4$) and initialised all (A_s, t_s) randomly and σ_s large. We first used only 2 EM-iterations on the whole sequence to train the parameters. Fig. 3B shows the result in terms of the MAP segmentation. The σ_s's became smaller and two flow field segments are dominating, which correspond to the tree and the background. Then, after 8 iterations of the EM-algorithm, the σ_s's further decreased and three segments are detected (Fig. 3C), which correspond to the tree, the flower garden and back branches, and the house and background. Note the smoothness of segmentation achieved based on the spatio-temporal coupling. Fig. 3D also displays the probabilistic segmentation: The forward filtered belief $\alpha(S^t)$ (in the 8^{th} EM iteration) for the four different segments $s = 1, .., 4$ are displayed. Compared to the MAP segmentation the probabilistic segmentation is smoother and captures more detailed information.

The last experiment tests a pure online segmentation using only forward filtering and an online EM which adapts the parameters at each time step. This targets towards real-time segmentation that does not rely on batch processing. Since in the flower garden sequence the local flow field measurement is rather precise we pre-specified a low variance $\sigma_s = 0.25$ for all s. We allowed for 3 segments ($K = 3$)

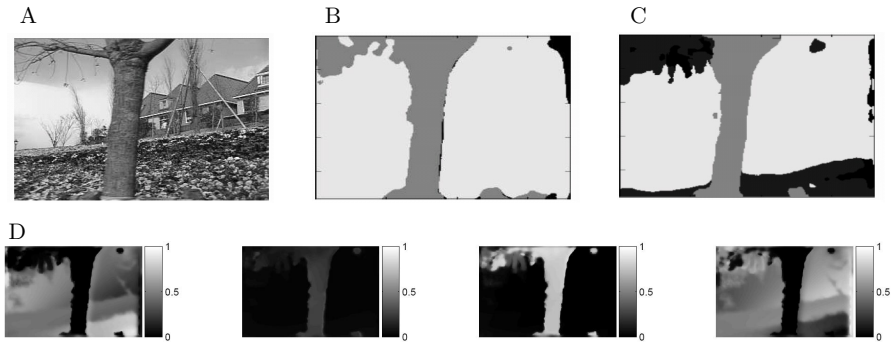


Figure 3: Flower Garden test using two-filter inference.

$t = 1$ $t = 2$ $t = 3$ $t = 4$ $t = 10$ $t = 20$ $t = 29$

Figure 4: Flower Garden test using an online forward filter and learning

and initialised one of them to zero $A_s = 0, t_s = 0$ (a prior that there are regions of low velocity) and the other two randomly. Fig. 4 displays the MAP segmentation for the online EM-filter after $t = 1, 2, 3, 4, 10, 20, 29$ time steps. In the first iteration the segmentation already distinguishes between the tree stem, the flower garden and branches, and the house and background. Initially this segmentation is rather noisy. During the online EM filter the segmentation becomes refined and also spatially more smooth due to the spatio-temporal priors that are build up during filtering.

5 Conclusion

Our approach focuses on exploiting spatio-temporal coherence in the flow field and segmentation evolution by formulating a Dynamic Bayesian Network framework as a basis for online filtering and inference over an image sequence. The core ingredients are the particular assumptions implied by our transition priors of the flow field (9) and the segmentation (10), and the efficient inference technique based on propagating factored beliefs over the flow field and segmentation forward and backward. Further, using the Student's t-distributions (4, 7) increases the robustness against outliers.

Both experiments have shown how the algorithm can extract a smooth segmentation of a sequence of images starting with a random initialisation of the segments' flow field and variance parameters. The smoothness is a result of the assumed transition prior rather than an additional MRF prior. This reduces the computational costs *per time step* considerably and is particularly interesting in view of online filtering problems as demonstrated by the last example. Here, an online parameter adaptation (online EM) can be based only on forward propagated beliefs with rather small computational cost per time step.

The second example also demonstrated the interesting effect of adapting the

variance associated to each segment. Large initialised σ_S's have the effect that all segments q_s try to learn similar global flow patterns. The resulting flat segmentation posterior has only minor influence on the flow field estimation. During the EM the σ_S's decrease (to increase the overall data likelihood), the segments start to specialise on certain spatial regions and the segmentation becomes more and more detailed.

Acknowledgements

Marc Toussaint was supported by the German Research Foundation (DFG), Emmy Noether fellowship TO 409/1-3.

References

[1] M.J. Black and P. Anandan. Robust dynamic motion estimation over time. In *CVPR*, pages 296–302, 1991.

[2] P. Burgi, A.L.Yuille, and N.M. Grzywacz. Probabilistic motion estimation based on temporal coherence. *Neural Computation*, 12(8):1839–1867, 2000.

[3] R. Dupont, O. Juan, and R. Keriven. Robust segmentation of hidden layers in video sequences. *IEEE ICPR*, 3:75–78, 2006.

[4] F. Heitz and P. Bouthemy. Multimodal estimation of discontinuous optical flow using markov random fields. *PAMI*, 15(12):1217–1232, 1993.

[5] K.P. Lim, A. Das, and M.N. Chong. Estimation of occlusion and dense motion fields in a bidirectional bayesian framework. *PAMI*, 24(5):712–718, 2002.

[6] Kevin Murphy and Yair Weiss. The factored frontier algorithm for approximate inference in DBNs. In *Proc. of the 17th Conf. on Uncertainty in Artificial Intelligence (UAI 2001)*, pages 378–385, 2001.

[7] S. Roth and M.J. Black. On the spatial statistics of optical flow. In *ICCV*, pages 42–49, 2005.

[8] A. A. Stocker and E. P. Simoncelli. Noise characteristics and prior expectations in human visual speed perception. *Nature Neuroscience*, 9(4):578–585, 2006.

[9] N. Vasconcelos and A. Lippmann. Empirical bayesian motion segmentation. *PAMI*, 23:217–221, 2001.

[10] Y. Wang, K.-F. Loe, T. Tan, and J.-K. Wu. Spatiotemporal video segmentation based on graphical models. *IEEE Trans. on Image Processing*, 14:937–947, 2005.

[11] Y. Weiss and E. H. Adelson. A unified mixture framework for motion segmentation: incorporating spatial coherence and estimating the numbers of models. *CVPR*, pages 321–326, 1996.

[12] V. Willert, J. Eggert, J. Adamy, and E. Körner. Non-gaussian velocity distributions integrated over space, time, and scales. *IEEE Transactions on Systems, Man and Cybernetics - Part B*, 36(3):482–493, 2006.

[13] K. Y. Wong, L. Ye, and M. E. Spetsakis. EM clustering of incomplete data applied to motion segmentation. *BMVC*, pages 237–246, 2004.

Layered Active Contours for Tracking

Gallagher Pryor, Patricio Vela, Tauseef ur Rehman, and Allen Tannenbaum
Georgia Institute of Technology, Atlanta GA, USA

Abstract

This paper deals with the task of object tracking in the presence of occlusions and clutter by fitting a layered appearance model to data. Four major problems must be overcome: (1) the association of each pixel to a particular layer (layer segmentation), (2) the determination of layer support, (3) the determination of layer appearance, and (4) determination of layer location. Tao, Sawhney, and Kumar successfully proposed a generalized expectation maximization algorithm solving these problems by directly inferring masks representing layer segmentation in conjunction with a deforming elliptical shape prior defining layer support. We extend their work with the introduction of active contours: instead of directly inferring these masks, we evolve a series of curves to obtain a layer segmentation. These curves provide a natural shape prior by constraining segmentations to a family of curves local to layer supports and allow for non-rigid layer deformations through the prediction of unobserved appearance information during inference. A benefit of this extension is the ability to track through massive occlusions and clutter, as demonstrated on a series of difficult real-world video sequences.

1 Introduction

Over the past few years, layered representations have been shown to be powerful tools for motion analysis and tracking. Sometimes referred to as motion segmentation in the literature, fitting a layered representation to imagery consists of segmenting video into regions (layers) containing homogeneous motion (from a given family of motions, e.g. translation), tracking those layers throughout the video, and building an appearance model for each layer. To solve the tracking problem within this framework, these segmentations, motions, and appearances are evolved in a causal fashion to fit imagery from a video stream.

Critical to the success of these algorithms are the constraints placed upon the layer, or motion, segmentations they generate. As such, many different classes of these constraints have been posed within the literature. For instance, layered segmentations performed via clustering have been constrained to meet certain size and smoothness criteria [12, 1]. In [6], similar constraints were proposed in conjunction with a constellation of parts model. In [14] segmentations were constrained to have short, smooth boundaries via graph cuts. Other segmentation constraints have been proposed that utilize a model-based approach involving approximate inference via *generalized expectation maximization* [5, 9, 13, 15]. In [4], a parameterized polybone skeleton shape prior is employed to constrain layered segmentations. Dai, Zheng, and Li [3] utilized a shape constraint specific to the problem of pedestrian detection and tracking. Toyama and Blake proposed a shape constraint facilitated by a learned metric [10]. Tao, Sawhney, and Kumar [8, 17] proposed the use of

an elliptical, probabilistic shape constraint, the most general of the proposed model-based constraints.

Contribution. In our work, we propose a tracking algorithm which extends the framework proposed by Tao, Sawhney, and Kumar. Instead of representing layer segmentations as probabilistic masks (or explicit models as in other works) we propose the use of active contours in their place. We believe that our approach gives four major advantages:

1. The curve representation enforces a general shape prior capable of handling complex shapes, but is still suitable for general tracking. This prior is achieved by constraining layer segmentations to the set of curves local to layers' physical shapes, or support.

2. The use of curves naturally gives rise to a method of handling non-rigid layer deformations through the prediction of unknown layer appearance information beyond the current layer support.

3. The resulting tracker can handle massive occlusions over long periods of time and can track through clutter and other similar targets due to the accurate layer segmentation made possible through **1** and **2**.

4. Unlike other segmentation algorithms such as [7, 14], the method is *causal* and maintains *explicit track points* based on the physical shape of the layers, not the segmentation itself.

We present results on seven video sequences that show this extension significantly improves segmentation and tracking performance over the original tracking algorithm.

Outline. In Section 2, we describe the layered active contour model and accompanying dynamic model. In Section 3, we describe the active layered tracking algorithm. In Section 4, we present results of tracking using a baseline tracker (a version of that proposed in [8]) and the proposed tracker. Finally, in Section 5 we give conclusions followed by directions for future work.

2 The Layered Active Contour Model

We define an instance of a layered representation $\Sigma = (\mathbf{C}, \mathbf{M}, \mathbf{G}, \mathbf{F}) = (\{C^i, M^i\}_{i=1}^{l-1}, \{\vec{g}^i, f^i\}_{i=0}^{l-1})$ that parameterizes the following family of imagery defined by the function F,

$$F(\vec{r}) = \begin{cases} f^k(\vec{r} - \vec{g}^k), & \vec{r} \in C_{in}^k \text{ and } \vec{r} - \vec{g}^k \in M_{in}^k \\ f^0(\vec{r} - \vec{g}^0), & \text{otherwise} \end{cases} \quad (1)$$

where:

- $l \in Z^+$ is defined as the number of layers utilized within the model, $k \in \{1, 2..., l-1\}$ and,
- $F : \Omega \rightarrow R$ where $\Omega \subset R^2$.

The data associated with each layer is:

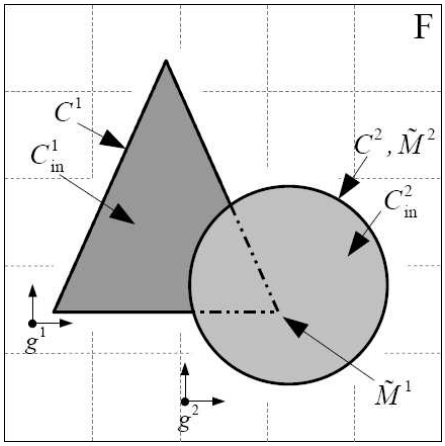


Figure 1: One parameterization of the layered model with two layers: a triangle f^1 occluded by a circle f^2 as indicated by the relative configuration of the curves C^1, C^2 (describing the visible region of each f) and $\tilde{M}^1, \tilde{M}^2$ (describing the support of each f).

- $\vec{g} \in \Omega$, a layer translation.
- Each C^k is a closed curve in Ω and each C^k_{in} is taken to be the set of all points enclosed by C^k. We enforce the constraint that the region enclosed by each curve is mutually exclusive of all other regions defined by their respective curves.
- Each M^k, like C^k, is a closed curve and, similarly, each M^k_{in} is taken to be the set of all points enclosed by each M^k. Unlike C, the curves M are not considered to be mutually exclusive since they each lie in different coordinate systems.
- Each $f^k : \Omega \rightarrow R$ represents the appearance of the k^{th} layer in the model; f^0 is assumed to be the appearance of the background of the model.

The function F describes all imagery composed of l independently translated 2D signals, each occluding one another. Figure 1 illustrates one member of this family, an image of a triangle (layer one) occluded by a circle (layer two). The 2D appearances of these layers are given by f^1 and f^2 with their respective shapes defined by M^1 and M^2. These objects are translated by the offsets $\vec{g}^1$ and $\vec{g}^2$ (resulting in each $\tilde{M}$), respectively, to make up the final output F. The curves C^1 and C^2 mark the boundary of the visible regions of each appearance function in F while $\tilde{M}^1$ and $\tilde{M}^2$ give the total support or shape of each layer. Note how the curve C^1 stops short of the circle, yet $\tilde{M}^1$ continues inward to completely describe the triangle, which indicates an occlusion.

In contrast to the model of [7], the proposed model utilizes curves instead of masks to describe layer segmentation and support. Curves have been proven to be particularly suited to the problem of segmentation while their evolution has been shown to yield sensible curve deformations making them ideal for facilitating flexible layers. Thus, through the introduction of the curves **C**, we leverage a proven solution to the subproblem of segmentation within layered decomposition, and through the introduction of the curves **M**, we constrain the curves **C** to a well-studied and sensible family of shapes. This approach also leads us to develop a completely maximum likelihood solution. In contrast to

generalized expectation maximization, our method does not rely on the maintenance of distributions over time.

2.1 Dynamic Layered Model

Using the function F defined above as an observation model, we define the likelihood of an observed set of images as $P(\mathbf{I}|\{\Sigma^t\}_{t=1}^T)$, where $\mathbf{I}$ is defined as $\{I^t\}_{t=1}^T$, a set of frames in a video sequence. We make a Markov assumption on the states Σ^t and consider each I^t to be conditionally independent given Σ^t, allowing us to factor the distribution,

$$P(\mathbf{I}|\{\Sigma^t\}_{t=1}^T) = \left[\prod_{t=2}^{T} P(I^t|\Sigma^t)P(\Sigma^t|\Sigma^{t-1})\right] P(I^1|\Sigma^1)P(\Sigma^1), \tag{2}$$

where $P(I^t|\Sigma^t)$ is taken to be the observation model,

$$P(I^t|\Sigma^t) = \frac{1}{Z}\exp\left[-\frac{1}{2\sigma}\int_\Omega (F(\vec{r}) - I(\vec{r}))^2 d\vec{r}\right], \tag{3}$$

and $P(\Sigma^t|\Sigma^{t-1})$ is taken to be the state dynamics model,

$$P(\Sigma^t|\Sigma^{t-1}) = P(\mathbf{G}^t|\mathbf{G}^{t-1}) = \prod_{i=0}^{l-1} \frac{\exp\left[-\frac{1}{2\sigma_g}(\vec{g}^{i,t} - \vec{g}^{i,t-1})^T(\vec{g}^{i,t} - \vec{g}^{i,t-1})\right]}{2\pi\sigma_g}, \tag{4}$$

To ensure that the posterior distribution $P(g^{i,t})$ remains Gaussian, we assume that $P(\vec{g}^{i,1}) = \delta(\vec{r} - \vec{g}^{i,1})$. We also assume that each f^i and M^i are fixed, independent of time.

3 Tracking via the Layered Model

To perform tracking via inference in the given probabilistic model, one must select an initial parameterization of F, or model state Σ^1, and evolve that state so as to maximize its likelihood given a series of observations $\mathbf{I}$. To do so, we treat this task as a filtering problem, assuming a causal series of observations, and adjusting each parameter set $\{\vec{g}^{i,t}, C^{i,t}\}_{i=0}^{l-1}$ in turn while recursively updating each $\{f\}_{i=0}^{l-1}$ accordingly with the new imagery.

3.1 Initialization

For this work, we assume a given initial segmentation as sufficient information to seed the algorithm. An initial segmentation consists of $l-1$ disjoint curves representing the initial visible regions of each layer to be tracked and an accompanying input image I^1. Each curve defines the state information $\mathbf{C}$ and $\mathbf{M}$ and the image information enclosed by each curve defines each $\mathbf{F}$. The state information $\mathbf{G}$ is defined to be zero, initially, essentially fixing a series of independent coordinate systems for each layer.

3.2 Tracking

Upon the arrival of each new observation I^t, we update the layered approximation to maximize the log likelihood of the current observation given the previous state. In doing

so, we also solve for the track point of each layer. Using the definitions we have presented up until now, this process reduces to the minimization of the following energy with respect to $\mathbf{G}, \mathbf{C}$, and $\mathbf{F}$,

$$E(\mathbf{G},\mathbf{C},\mathbf{F}) = \frac{1}{2\sigma}\int_{\Omega}(F(\vec{r}) - I^t(\vec{r}))^2 d\vec{r} + \sum_{i=0}^{l-1}\frac{1}{2\sigma_g}(\vec{g}^{i,t} - \vec{g}^{i,t-1})^T(\vec{g}^{i,t} - \vec{g}^{i,t-1}), \quad (5)$$

where the first term of this energy forces the approximation F to match the current observation I^t while the second term enforces a penalty on large track signal jumps from frame to frame weighted by the model parameter σ_g.

In the derivation that follows, the updates on each parameter set $\mathbf{G}, \mathbf{C}, \mathbf{F}$ of F are derived on a layer by layer basis to minimize this energy. In the cases of $\mathbf{G}$ and $\mathbf{F}$, a closed-form minimization is performed, but in the case of $\mathbf{C}$ and $\mathbf{M}$, which are curves, we apply gradient descent.

3.2.1 Solving for $\vec{g}^{i,t}$

Due to the fact that we make no assumptions about the functions f (smoothness, etc), the landscape of E is possibly complex with many local minima with respect to each $\vec{g}^{i,t}$. Additionally, depending on the value of the externally defined σ_g, the minimizer $\vec{g}^{i,t}$ will always lie within a set distance of $\vec{g}^{i,t-1} \in \Omega$, defining a region Ω'. Thus, we minimize E with respect to $\vec{g}^{i,t}$ through direct search over Ω',

$$\vec{g}^{i,t} = \underset{\vec{g}^{i,t}}{\operatorname{argmin}}\left[\int_{\Omega}(F(\vec{r},\tau) - I(\vec{r},\tau))^2 d\vec{r} + \frac{1}{2\sigma_g}(\vec{g}^{i,t} - \vec{g}^{i,t-1})^T(\vec{g}^{i,t} - \vec{g}^{i,t-1})\right] \quad (6)$$

3.2.2 Solving for $C^{i,t}$

To solve for $C^{i,t}$ we perform gradient descent starting from the translated curve $\tilde{M}^i$ as an initial guess. In doing so, we constrain the curve $C^{i,t}$ to lie in the local neighborhood of $\tilde{M}^i$, enforcing the constraint that the shape of layer i's visible region is within a set distance of the underlying support of that layer. The size of this neighborhood is determined by the number of gradient descent iterations we perform on $C^{i,t}$, n. Additionally, by evolving $C^{i,t}$ according to a local flow, we simulate the event of an occlusion or shape change of a layer: starting from the initial layer support M^i, $C^{i,t}$ must move pixel by pixel either inwards or outwards along its normals as exactly would be seen during an occlusion or shape change of the layer. The mirroring of this physical process in our inference step, we believe, justifies our technique of providing a shape constraint within our model.

To perform gradient descent on $C^{i,t}$, we find an energy-minimizing flow on $C^{i,t}$ by taking the first variation of the energy E with respect to $C^{i,t}$. For the sake of exposition, we begin with the simple case when $l = 2$ where only one curve is present within the model. In this case, the energy E simplifies down to a competition between two quadratic terms:

$$E(\mathbf{C}) = \int_{C^{1,t}_{in}}(f^1(\vec{r} - \vec{g}^{1,t}) - I^t(\vec{r}))^2 d\vec{r} + \int_{C^{1,t}_{out}}(f^0(\vec{r} - \vec{g}^{0,t}) - I(\vec{r}))^2 d\vec{r} \quad (7)$$

Applying Green's theorem and taking the first variation of this simplified energy with respect to C^1 gives the following flow for the curve,

$$\frac{\partial C^{1,t}(\tau)}{\partial \tau} = \vec{N}[(f^1(\vec{r} + \vec{g}^{1,t}) - I^t(\vec{r}))^2 - (f^0(\vec{r} + \vec{g}^{0,t})) - I^t(\vec{r}))^2], \quad (8)$$

where τ is the typical artificial time parameter used in conjunction with gradient descent and $\vec{N}$ is the normal of the curve $C^{1,t}$. The flow (8) is related to that proposed by Chan and Vese [2], but is unique with its more specific appearance model. Heuristically, this flow expands $C^{1,t}$ along its normal wherever I^t is more similar to layer one than layer zero and vice versa. Note that no regularizing terms are present in this flow: we do not expect the visible region of an object to have any special properties, though it makes sense to assume so for the layer support M^i.

In the general case where $l > 2$, the energy E simplifies into a competition between $l-1$ curves:

$$E(\mathbf{C}) = \sum_{i=1}^{l-1} \int_{C_{in}^{i,t}} (f^i(\vec{r} - \vec{g}^{i,t}) - I^t(\vec{r}))^2 d\vec{r} + \int_{\bigcap_{i=1}^{l-1} C_{out}^{i,t}} (f^0(\vec{r} - \vec{g}^{0,t}) - I^t(\vec{r}))^2 d\vec{r}, \quad (9)$$

where $\bigcap_{i=1}^{l-1} C_{out}^{i,t}$, defines the region(s) in Ω exterior to all curves in the model. To determine the minimizing curves $\mathbf{C}$, the variation of each curve must be taken and gradient descent carried out while enforcing the constraint that all $\mathbf{C}$ must be mutually disjoint. Several researchers have studied this problem and many solutions exist to satisfy this constraint including the addition of an additional energy term with Lagrange multiplier [16] or utilizing a modified level set representation [11]. However, these solutions result in the same type of flow as in the simplified, two layer case where all curves move along their normals, competing against one another according to a quadratic matching term with I^t.

3.2.3 Solving for f^i

To solve for an updated version of each f^i, we take the derivative of the energy, set to zero, and solve for f^i. Note that according to our dynamical model, f^i is assumed to be independent of time. Thus, the error minimizing value for f^i is computed to be a time average of all observed data within the bounds of each $\{C^{i,k}\}_{k=1}^t$,

$$f^i(\vec{r}) = \frac{\sum_{k=1}^t I^k(\vec{r} - \vec{g}^{i,k}) H^{i,k}(\vec{r} - \vec{g}^{i,k})}{\sum_{k=1}^t H^{i,k}(\vec{r} - \vec{g}^{i,k})}, \quad (10)$$

where we define $H^{i,k}$ to be the Heaviside function indicating the enclosed region of the curve $C^{i,k}$,

$$H^{i,k}(\vec{r}) = \begin{cases} 1, & \vec{r} \in C_{in}^{i,k} \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

3.3 Flexible Layers via Appearance Prediction

Due to the construction of the layered observation model where layers occlude one another and have flexible support, it may be impossible to observe values for the appearance functions $\mathbf{F}$ in the given data. For instance, though each observation I^t has support over Ω, the layer appearance functions $\mathbf{F}$ initially contain information only on the supports defined by $\mathbf{M}$ which are typically a strict subset of Ω. As such, any value of f^i beyond this support has yet to be observed and is considered unknown information. Information in the background may also be unknown due to occluding foreground layers.

Unfortunately, while solving for $C^{i,t}$, an estimate of this unknown information is necessary for inference as it is possible for $C^{i,t}$ to grow outward beyond M^i (in the case of

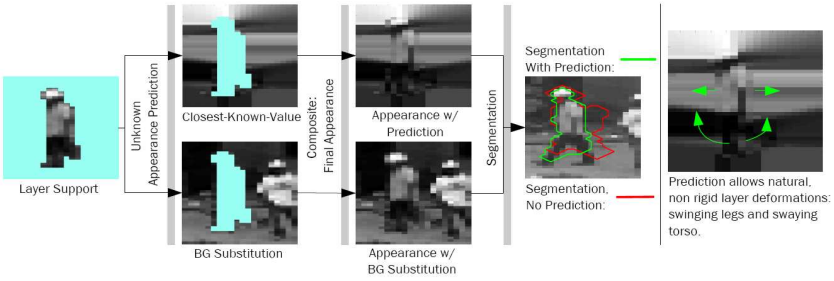


Figure 2: Flexible layers are facilitated by closest-known-value prediction of unknown appearance that allow natural non-rigid layer deformations including the swinging of the worker's legs and torso movement. This leads to improved layer segmentation over using no prediction.

layer shape change) or away from occluded background regions (in the case where a layer translates away from its initial position). We handle this problem while at the same time implicitly imposing a shape constraint on the model's appearances and supports by adhering to the closest-known-value principle: we define each unknown value $f^i(\vec{r})$ to be equal to the closest known, or observed, value f^i to $\vec{r}$,

$$f^{i\prime}(\vec{r}) = \begin{cases} f^i(\vec{r}), & \sum_{k=1}^{t} H^{i,k}(\vec{r}) > 0 \\ f^i(\text{argmin}_{\vec{x}\in M^i} \|\vec{x}-\vec{r}\|), & \text{otherwise} \end{cases} \tag{12}$$

This technique is often used in the active contour community to facilitate what are known as velocity or data extensions whenever information is required off of an interface to evolve a higher-dimensional embedding function. We use this technique, here, to enforce two assumptions: **(1)** Initially unobserved information due to occlusion is expected to be equal to its closest known neighbor and is therefore geometrically consistent and **(2)** when layers change shape, we expect edge appearance information to be attached to the edges of the shape. Through this prediction technique, many common shape and appearance changes can be handled (see Figure 2).

4 Results

4.1 Experimental Setup

For comparison purposes, we implemented a version of the tracker proposed by Tao, Sawhney, and Kumar as the baseline tracker. We then extended this implementation so as to include the active contour model we have presented, leading to a change regarding *only* the layer segmentation calculation and unknown appearance information. All other inference steps and model parameters remained fixed. Because our focus is on tracking, we used a consistent hand segmentation for initialization of both algorithms, though both can be initialized automatically with no drawbacks [8].

All experiments were carried out by running both the trackers on seven 128x128 video sequences. On average, each video sequence consisted of 300 frames or 10 seconds of

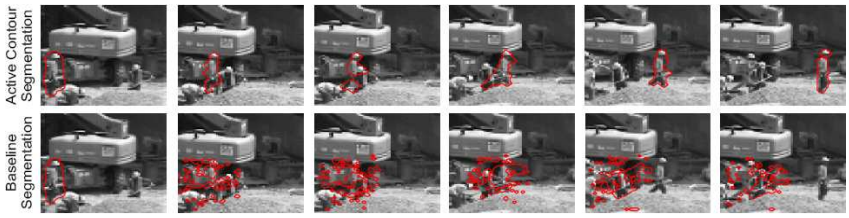


Figure 3: Tracking and segmentation results on video 5 from Table 1. The red line marks the layer segmentation curve (C^1) in the first row and the 50% probability isocontour of the layer segmentation in the second row. The active contour tracker maintains an accurate layer segmentation and track signal in the presence of clutter and similar moving targets.

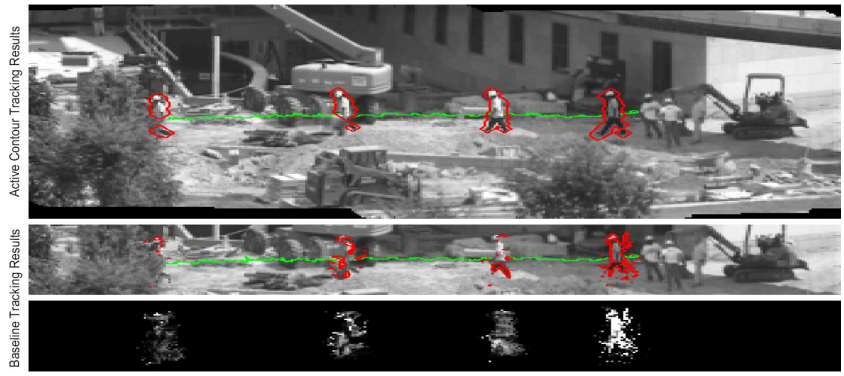


Figure 4: Example of tracking a construction worker with a left-panning camera and changing lighting conditions. Here, the final appearance, or panorama, of the background layer is overlaid with layer one's visible region indicated by the curve C^1 (red) from frames 12, 109, 228, and 387 and track signal (green). The bottom two rows give the baseline tracker's track signal, the red contour indicating the 50% probability isocontour of the layer segmentation and the last row indicating the likelihood of each pixel belonging to the foreground layer. The active contour tracker maintains a correct layer segmentation and an accurate track signal.

imagery. To generate ground truth track signals for comparison, a hand-picked series of "correct" track points were chosen on each frame for each layer on each video. Performance results were generated by comparing the tracker's output track signals for each layer, $q^t_{x,y}$, against the hand-picked track points, or ground truth signal $r^t_{x,y}$.

4.2 Analysis

A quantitative comparison between the layered active contour tracker and the baseline tracker is given in Table 1 using the performance metrics described above on a single foreground layer's track signal. Unlike the baseline tracker, the active contour tracker maintained track through all seven sequences. In the cases both trackers maintained track, the active contour tracked with less average error.

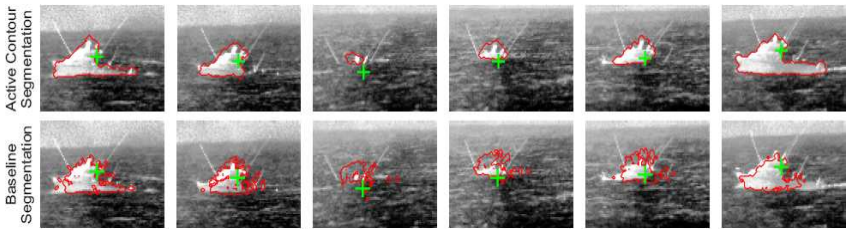


Figure 5: Tracking and segmentation results of tracking a boat in the open ocean. As the boat sails perpendicular to the line of sight, it is almost totally obscured by a passing swell of water for approximately 100 frames. The red line marks the layer segmentation curve (C^1) in the second row and the 50% probability isocontour of the layer segmentation in the third row. The green cross marks the track point that, for both trackers, stays on target. The active contour maintains a correct layer segmentation even through the massive occlusion.

Tracker Performance:
Layered Active Contour vs Baseline Tracker

Videos								
Active Contour	**%**	100%	100%	100%	100%	100%	100%	100%
	ε	1.25	0.68	0.81	1.66	1.97	0.76	1.03
Baseline	**%**	78%	100%	100%	100%	49%	97%	100%
	ε	**LT**	0.71	0.85	3.97	**LT**	0.9	1.23

Table 1: Percent of time on track (**%**) and mean track error (ε) of the layered active contour model versus the baseline layered tracker. **LT** Indicates total track loss.

The most clear advantage the active contour had was on video 5. Here, a track subject moves amongst a great deal of clutter and dust, with similar targets in proximity to the tracking subject (Figure 3). The more accurate active contour segmentation (as opposed to a mask) allowed the tracker to avoid learning an incorrect layer appearance and thus handle the ambiguous case of crossing workers. In each of the other videos, the active contour maintains an accurate segmentation and track signal (Figures 4 and 5).

5 Conclusions

In this paper, we have developed a tracker based upon a layered active contour model as an extenstion to [8]. In place of representing layer segmentation as a series of masks, we utilized curves that are evolved to fit data. We showed that as a result of this extension, a general shape prior is introduced into the model and a natural way of handling non-rigid layer deformations is introduced through the prediction of unknown appearance information. Finally, we presented results on several difficult video sequences demonstrating the benefits of this addition to the model.

References

[1] M. Allen, M. K. Titsias, and C. K. I. Williams. Fast learning of sprites using invariant features. In *Proc. British Machine Vision Conference*, 2005.

[2] T. Chan and L. Vese. Active contours without edges. *IEEE Trans. Image Processing*, pages 266–277, February 2001.

[3] C. Dai, Y. Zheng, and X. Li. Layered representation for pedestrian detection and tracking in infrared imagery. In *Proc. IEEE CVPR*, volume 3, page 13, June 2005.

[4] A. D. Jepson, D. J. Fleet, and M.J. Black. A layered motion representation with occlusion and compact spatial support. In *Proc ECCV*, pages 692–706, 2002.

[5] N. Jojic and B. Frey. Learning flexible sprites in video layers. In *Proc. IEEE CVPR*, volume 1, pages 199–206, Maui, HI., 2001.

[6] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Learning layered motion segmentation of video. In *Proc. IEEE ICCV*, volume 1, pages 33–40, 2005.

[7] Nikos Paragios and Rachid Deriche. Geodesic active regions and level set methods for motion estimation and tracking. *Comput. Vis. Image Underst.*, 97(3):259–282, 2005.

[8] H. Tao, H. S. Sawhney, and R. Kumar. Object tracking with Bayesian estimation of dynamic layer representations. *IEEE Trans. PAMI*, 24:75–89, 2002.

[9] P. H. S. Torr, R. Szeliski, and P. Anandan. An integrated Bayesian approach to layer extraction from image sequences. *IEEE Trans. PAMI*, 23:297–303, March 2001.

[10] K. Toyama and A. Blake. Probabilistic tracking with exemplars in a metric space. *Int. J. Computer Vision*, pages 9–19, 2002.

[11] L. A. Vese and T. F. Chan. A multiphase level set framework for image segmentation using the Mumford and Shah model. Technical Report 01-25, University of California, Los Angeles, September 2001.

[12] J. Y. A. Wang and E. H. Adelson. Layered representation for motion analysis. In *Proc. IEEE CVPR*, pages 361–366, New York, June 1993.

[13] C. K. I. Williams and M. K. Titsias. Greedy learning of multiple objects in images using robust statistics and factorial learning. *Neural Computation*, 16:1039–1062, 2004.

[14] J. Xiao and M. Shah. Motion layer extraction in the presence of occlusion using graph cuts. *IEEE Trans. PAMI*, 27:1644–1659, October 2005.

[15] H. Yalcin, M.J. Black, and R. Fablet. The dense estimation of motion and appearance in layers. In *Proc. IEEE CVPR*, page 165, June 2004.

[16] H.-K. Zhao, T. Chan, M. Merriman, and S. Osher. A variational level set approach to multiphase motion. *Journal of Computational Physics*, 127:179–195, August 1996.

[17] Y. Zhou and H. Tao. A background layer model for object tracking through occlusion. In *Proc. IEEE ICCV*, volume 2, pages 1079–1085, October 2003.

Automatic Player Detection, Labeling and Tracking in Broadcast Soccer Video

Jia Liu[1,2], Xiaofeng Tong[3], Wenlong Li[3], Tao Wang[3], Yimin Zhang[3], Hongqi Wang[1], Bo Yang[4], Lifeng Sun[4], Shiqiang Yang[4]
[1]Institute of Electronics, Chinese Academy of Sciences, Beijing 100080, China
[2]Graduate University of Chinese Academy of Sciences, Beijing 100039, China
[3]Intel China Research Center, Beijing 100080, China
[4]DCST, Tsinghua University, Beijing 100080, China
liujia_iecas@hotmail.com, {xiaofeng.tong, wenlong.li, tao.wang, yimin.zhang}@ intel.com
wiecas@sina.com, yangbo1984@gmail.com, {sunlf, yangshq}@tsinghua.edu.cn

Abstract

Automatic player detection, labeling and tracking in broadcast soccer video are significant while quite challenging tasks. In this paper, we present a solution to perform automatic multiple player detection, unsupervised labeling and efficient tracking. Players' position and scale are determined by a boosting based detector. Players' appearance models are unsupervised learned from hundreds of samples automatically collected by detection. Thereafter, these models can be utilized for player labeling (Team A, Team B and Referee). Player tracking is achieved by Markov Chain Monte Carlo (MCMC) data association. Some data driven dynamics are proposed to improve the Markov chain's efficiency. The testing results on FIFA World Cup 2006 video demonstrate that our method can reach high detection and labeling precision, and reliably tracking in cases of scenes such as multiple player occlusion, moderate camera motion and pose variation.

1 Introduction

Automatic player localization, labeling and tracking is critical for team tactics, player activity analysis and enjoyment in broadcast sports videos. It is quite challenging due to many difficulties such as player-to-player occlusion, similar player appearance, varying number of players, abrupt camera motion, various noises, video blur, etc.

Many algorithms have been presented to deal with the multiple target tracking problem, such as particle filter [1][2], joint probabilistic data association filter (JPDAF) [3], multiple hypothesis tracking (MHT) [4], MCMC data association [5][6] and track linking [7][8][9]. Several researchers also investigated the specific problem of labeling and tracking of players in sports video [2][10][11]. In [10], a clustering based trajectory matching method was proposed to solve the tracking of players in soccer video. In their work, labeling of individuals was achieved by supervised classification. [11] built a track graph, and took the tracking problem as inference in a Bayesian network. In both

of these two works, a multi-camera system was used to get a stationary, high-resolution and wide-field view of soccer game. This setting ensured a reliable background subtraction can be obtained. In our application, the camera is not fixed, which results in moving background. Thus, we need robust and adaptive background modeling and effective object association technologies. In another aspect, unsupervised player labeling is preferred for its generalization ability.

In this paper, we propose a solution for player detection, labeling and tracking in broadcast soccer video. The system framework is illustrated in Figure 1. The whole procedure is a two-pass video scan. In the first scan, we (1) learn video dominant color via accumulated color histograms, and (2) unsupervised learn players' appearance models over hundreds of player samples collected by a boosted player detector. In the second scan, that is the testing procedure, we first use the dominant color for playfield segmentation and view-type classification. Then we apply a boosting player detector to localize players. Afterwards, the players are labeled as Team A, Team B or Referee with prior learned models. Finally, we perform data-driven MCMC association to generate players' trajectories, in which track length, label consistency and motion consistency are used as criterions for associating observations across frames.

The main contributions of our method are: (1) robust player detection achieved by background filtering and a boosted cascade detector; (2) unsupervised player appearance modeling, the referee can be identified in addition to two teams players without any manual labeling; (3) efficient global data association for player tracking, which solves the difficulties like tracker coalescence and correspondence maintaining after occlusions.

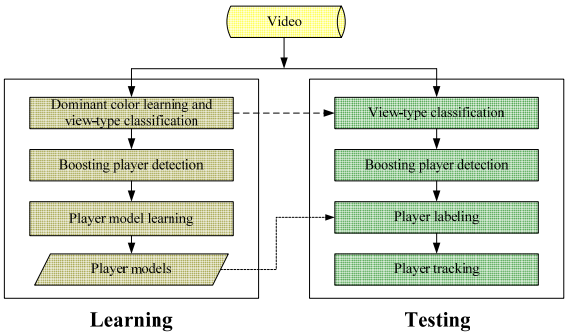


Figure 1: System framework

The paper is organized as follows: Section 2 describes the dominant color learning and view-type classification. Section 3 presents the boosting based player detection method. Player labeling and tracking algorithm are described in section 4 and 5. Experimental results are shown in section 6, and conclusions are made in section 7.

2 Dominant Color Learning and View-type Classification

We first learn the dominant color of background (corresponds to grass color of the playfield) by accumulating HSV color histograms. Then the playfield is extracted

through dominant color segmentation, morphological filtering and connect-component analysis. According to the size of playfield and non-field regions, we use a decision tree to classify each view into one of four pre-defined view types: global view, medium view, close-up, and out of view. The detailed algorithm is described in our previous work [12]. Only global views are fed into player detection module.

3 Player Detection

Player detection is achieved by running a boosted cascade of Haar features [13] on global views. We manually labeled about 6000 players as positive samples. These samples are carefully selected in order to partially capture the appearance variation of players caused by body articulate motion. Negative samples are background patches randomly cropped from soccer video images. Some training samples are shown in Figure 2. All these samples are properly scaled to a resolution of 32×64 pixels.

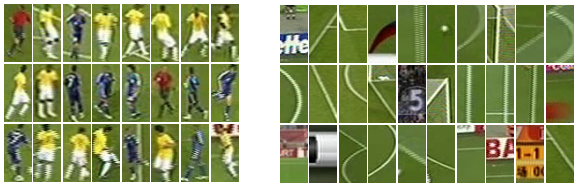


Figure 2: Some of positive and negative training samples used in training

A boosted cascade detector is then trained on this sample set. To accelerate the training process, we randomly select only a fraction $\varepsilon = 0.01$ of Haar features for each round training as in [14]. This process brings dramatic speed up with only small impact on the final detection performance.

In detection phase, playfield segmentation is first used to filter out the background regions. This process accelerates the detection and reduces false alarms. The detector is then scanned across the filtered image regions at multiple scales. Multiple detections will usually occur around each player after scanning the image. We merge adjacent detected rectangles and get final detections with proper scale and position.

4 Player Labeling

The task of player labeling is to distinguish players' identities (Team A, Team B, Referee). Since now, we use word "player" indiscriminately to represent both team players and referee when there is no confusion. At present, we do not consider the problem of labeling goalkeepers.

4.1 Player Appearance Modeling

To learn the player appearance model, we run the player detector on every 50 frames to collect training samples from the input video. About 500 frames are processed and approximately 1,500 player samples are extracted from a half game video (about 45 mins, 25 fps).

We utilize a *bag of feature* representation of players. This process is illustrated in

Figure 3. For each player sample, the background is subtracted by using the dominant color model firstly, and only the upper body region is used for training. A large pool of pixels is then collected from these regions and transformed into CIE-Luv color space. We estimate a Gaussian Mixture Model (GMM) with N components in the space by Expectation-Maximization (EM) [15] clustering. Centers of these components are named *prototypes*. The adjacent components with small center distance are then merged together. The resultant merged components are called *meta-prototypes*. Each player sample is then represented as a histogram by binning all pixels in upper body region into the corresponding meta-prototype.

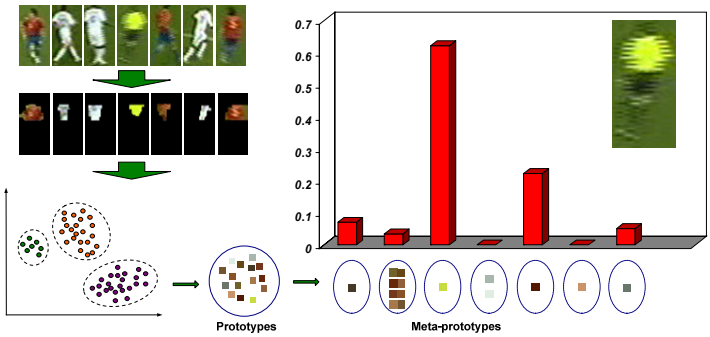


Figure 3: Bag of feature representation of players

Then we learn the appearance model for each identity (Team A, Team B, Referee). A naive clustering on player samples is not enough here. Actually, we manually labeled some samples collected from a real match video, and plotted the first two dimensions of PCA in feature space, as shown in Figure 4. It can be seen that referee samples do not clearly form a cluster in feature space due to their small quantity. A direct clustering may lead them to be absorbed into a nearby cluster. The existing of outliers will also incur an imprecise localization of clusters.

To deal with these problems, we first use EM clustering to estimate K clusters over the meta-prototype histogram of all player samples. K should be large enough to make sure that the referee samples form at least one cluster. Centers of these K clusters are named *sub-models*, which are denoted as $SM = \{sm_i, i = 1,2,...,K\}$. Then we merge adjacent clusters into four clusters by hierarchical clustering. Their centers are named *real-models*, and formally denoted as $RM = \{rm_i, i = 1,2,3,4\}$. A labeling function L assigns each real-model and sub-model exactly one label in a label set $LS = \{\text{Team A, Team B, Referee, Outlier}\}$. The two real-models with the largest size are labeled as Team A and Team B, as well as their corresponding sub-models. Then we compute a *minimum average distance* (MAD) from other two real-models to the team sub-models. Denote samples in a real-model as $s_i, i = 1,2,...,|rm|$. All the sub-models labeled as Team A or Team B are denoted as $TM = \{tm_j : tm_j \in SM \text{ and } L(tm_j) \in \{\text{Team A,Team B}\}, j = 1,2...\}$. The MAD is defined as:

$$MAD(rm) = \sum_{i=1}^{|rm|} \min_j (Bd(s_i, tm_j)) \Big/ |rm| \quad (1)$$

Where Bd is the Bhattacharyya distance defined as in [2]:

$$Bd(h_1,h_2)=\left[1-\sum_{n=1}^{N}\sqrt{h_1(n)h_2(n)}\right]^2 \quad (2)$$

The real-model with a larger MAD is labeled as Referee, and the other one is labeled as Outlier. All the sub-models labeled as Outlier are discarded and never used for future testing, i.e., We only maintain a set of sub-models take label from label set $LS=\{\text{Team A, Team B, Referee}\}$.

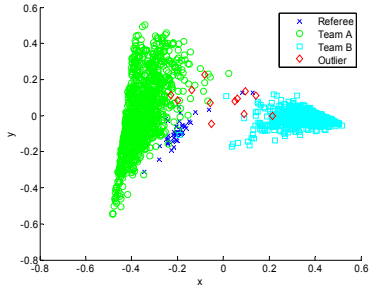


Figure 4: Scatter plot of first two dimensions of PCA in feature space

4.2 Player Labeling

In player labeling phase, each player sample is represented by its meta-prototype histogram, which is denoted as s. We calculate the Bhattacharyya distance between s and each sub-model sm as in formula (2). The sample is assigned the sub-model's label with the nearest distance (nearest neighbor).

5 Tracking

5.1 Problem Formulation

We formulate the tracking of multiple players as a data association problem. The whole detection and labeling result over a time period [1, T] is taken as the observation set, which is denote as $Z_{1:T}$. Let Z_t be the observations at time t, and Z_t^i be the i th observation at time t. To reduce the computational complexity, we define a neighborhood graph $G=(V,E)$ on the observation set, with each node in the graph represents a single observation, and edges defined between neighboring nodes. The neighboring of nodes is defined as:

$$N=\left\{(Z_{t1}^i,Z_{t2}^j):\left\|Z_{t1}^i-Z_{t2}^j\right\|\le\|t1-t2\|\times v_{\max}\ \text{ and }\ \|t1-t2\|\le T_{\max}\right\} \quad (3)$$

where $v_{\max}$ is the maximum speed of targets and $T_{\max}$ is the maximum time interval of consecutive observations in a track. A partition of such neighborhood graph can be represented as $\omega=\{\tau_0,\tau_1,\tau_2,...,\tau_K\}$, where τ_0 denotes false alarms and τ_k denotes the k th track. τ_k takes the form $\tau_k=\{\tau_k(t_1),\tau_k(t_2),...,\tau_k(t_{|\tau_k|})\}$, where $\tau_k(t_n)$ is an observation at time t_n. The global optimal association can be represented by a partition ω^* which maximizes a posterior of ω given observations:

$$\omega^*=\arg\max\left(P(\omega\mid Z)\right) \quad (4)$$

The posterior is formulated in a Gibbs distribution form as follows:

$$P(\omega \mid Z) = \frac{1}{C}\exp\left[\left(-\sum_{k=1}^{K} U(\tau_k) - \sum_{i \neq j} V(\tau_i, \tau_j)\right) \Big/ T\right] \tag{5}$$

where C is normalizing constant, T is temperature, U is a potential based on temporal compatibility within a single track, and V is a potential based on spatial compatibility between different tracks.

The potential U of a track τ_k is formulated as:

$$U(\tau_k) = U_{length}(\tau_k) + U_{label}(\tau_k) + U_{motion}(\tau_k) \tag{6}$$

with U_{length} penalizes short track, U_{label} favors consistent labeling, and U_{motion} penalizes inconsistent motion and long range association. Let $T_d(\tau_k)$ be the duration of τ_k, L_{th} be the expected length of tracks, U_{length} is defined as:

$$U_{length}(\tau_k) = \begin{cases} \alpha\left[1 - 2 \cdot \left(T_d(\tau_k) / L_{th}\right)^2\right], & \text{if } T_d(\tau_k) < L_{th} \\ -\alpha, & \text{if } T_d(\tau_k) \geq L_{th} \end{cases} \tag{7}$$

We define the label of a track to be the most frequently appeared observation label in the track. The label potential U_{label} is defined as:

$$U_{label}(\tau_k) = \beta \sum_{i=1}^{|\tau_k|} U_l(L(\tau_k(t_i)) \mid L(\tau_k)) \tag{8}$$

where L is the label of an observation or a track. U_l is defined as:

$$U_l = \begin{cases} -\varpi_s, & \text{if } L(\tau_k(t_i)) = L(\tau_k) \\ \varpi_d, & \text{if } L(\tau_k(t_i)) \neq L(\tau_k) \end{cases} \tag{9}$$

where ϖ_s and ϖ_d are positive potential. The motion potential U_{motion} is defined as:

$$\begin{aligned} U_{motion}(\tau_k) &= \sum_{i=1}^{|\tau_k|-1} U_m(\tau_k(t_{i+1}) \mid \tau_k(t_i)) \\ &= \sum_{i=1}^{|\tau_k|-1} \left\{ \varepsilon \cdot \left[\left\| \tau_k(t_{i+1}) - \tau_k(t_i) \right\| \Big/ \left[(t_{i+1} - t_i) \times v_{\max} \right] \right]^2 + \sigma \cdot \left[(t_{i+1} - t_i) / T_{\max} \right]^2 \right\} \end{aligned} \tag{10}$$

The potential term V penalizes spatial overlapping between tracks with same label:

$$V(\tau_i, \tau_j) = \begin{cases} \sum_{m=1}^{|\tau_i|} \sum_{n=1}^{|\tau_j|} \theta\rho(\tau_i(m), \tau_j(n)) & \text{if } L(\tau_i) = L(\tau_j) \\ 0 & \text{if } L(\tau_i) \neq L(\tau_j) \end{cases} \tag{11}$$

where ρ is the spatial overlap between two observation nodes.

5.2 MCMC Data Association

It is usually impossible to compute the global optimal solution for (5) analytically. We adopt MCMC strategy to explore the solution space, and estimate the optimal solution by a simulated annealing scheme. Here we use Metropolis—Hastings (MH) sampler [16] as the MCMC sampler. The design of moves in MH algorithm is crucial to the efficiency of Markov chain. Our move set M consists of seven types of moves (shown in Figure 5): $M = \{birth, death, extension, reduction, split, merge\}$. They are similar to those used in [6]. These moves are grouped into reversible pairs: *birth* / *death* pair, *extension* / *reduction* pair and *split* / *merge* pair. Now we describe these move pairs and deduce the proposal ratio for them. We denote current state as $\omega = (\tau_0, \tau_1, \tau_2, ..., \tau_K)$ and proposed state as ω' in each move. Proposal ratio for each move is denoted as R_m, $m \in M$.

(1) Birth/Death Move Pairs

The birth/death move consists of adding/removing a track from current state. For birth move, denote proposed state as $\omega' = (\tau'_0, \tau_1, \tau_2, ..., \tau_K, \tau'_{K+1})$. First, we select u.a.r

(uniform at random) a node $z_c \in \tau_0$ in ω, associate it to a new track τ'_{K+1}. An extension direction flag d is selected u.a.r from direction set *D={forward, backward}*. Then we start an association procedure: For each $z_i \in N^d(z_c)$, assign an association probability $p_d(z_i)$. Choose z_i with probability $p_d(z_i)$, associate it to track τ'_{K+1}, and set $z_c = z_i$. Continue the association with probability γ ($0 < \gamma < 1$). Repeat this procedure n times until an association is rejected or no observation can be added. For death move, denote proposed state as $\omega' = (\tau'_0, \tau_1, \tau_2, ..., \tau_{K-1})$. We select u.a.r a track τ_k in ω and remove it. The proposal ratios are given by:

$$R_{birth}(\omega \to \omega') = 2|\tau_0| \Big/ \left[(K+1) p_{end} \gamma^{n-1} \prod_{s=1}^{n} p_d(z_i^s) \right] \tag{12}$$

$$R_{death}(\omega \to \omega') = \frac{K\gamma^{(|\tau_K|-2)} \left\{ p_{end-forward} \prod_{s=1}^{(|\tau_K|-1)} p_{forward}(z_i^s) + p_{end-backward} \prod_{s=1}^{(|\tau_K|-1)} p_{backward}(z_i^s) \right\}}{2(|\tau_0| + |\tau_K|)} \tag{13}$$

where p_{end} is the probability of ending association, $p_{end-forward}$ and $p_{end-backward}$ are the probability of ending association at tail and head. The values of these probabilities are 1 if no observation can be added or $1-\gamma$ otherwise.

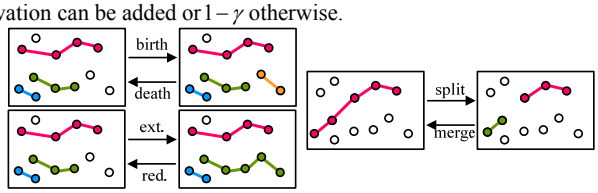


Figure 5: MCMC moves used in proposal

(2) Extension/Reduction Move Pairs

The extension/reduction move consists of extending/reducing a track in one direction. For extension move, denote proposed state as $\omega' = (\tau'_0, \tau_1, \tau_2, ..., \tau'_K)$. First, a track τ_k is selected u.a.r in ω and extension direction flag d is selected u.a.r from *D*. Then we repeat the same association procedure n times as in birth move on d. For reduction move, denote proposed state as $\omega' = (\tau'_0, \tau_1, \tau_2, ..., \tau'_K)$. First, a track τ_k is selected u.a.r in ω. Then a cutting index is selected u.a.r from {2, 3, …, $|\tau_k| - 1$}. A reduction direction flag d is selected u.a.r from *D*, and all the observations on this direction are removed from the track. The proposal ratios are given by:

$$R_{extension}(\omega \to \omega') = 1 \Big/ \left[(|\tau_K| + n - 2) p_{end} \gamma^{n-1} \prod_{s=1}^{n} p_d(z_i^s) \right] \tag{14}$$

$$R_{reduction}(\omega \to \omega') = (|\tau_k| - 2) p_{end} \gamma^{|\tau'_k| - |\tau_k| - 1} \prod_{s=1}^{|\tau'_k| - |\tau_k|} p_d(z_i^s) \tag{15}$$

(3) Split/Merge Move Pairs

The split/merge move consists of splitting a single track or merging two neighboring tracks. For a split move, denote proposed state as $\omega' = (\tau_0, \tau_1, \tau_2, ..., \tau'_K, \tau'_{K+1})$. First, we select u.a.r a track τ_k in ω. Then select a split point t_{split} with probability $p(\tau_k(t_{split}))$ and split the track into two new tracks. For a merge move, denote proposed state as $\omega' = (\tau_0, \tau_1, \tau_2, ..., \tau'_{K-1})$. The move select two tracks τ_i and τ_j with probability $p_{merge}(\tau_i, \tau_j)$ from all the pair candidates in ω, and merge them into a single track τ'_{K-1}. The proposal ratios are given by:

$$R_{split}(\omega \to \omega') = p_{merge} \cdot K \Big/ p(\tau_k(t_{split})) \tag{16}$$

$$R_{merge}(\omega \to \omega') = p(\tau'_{K-1}(t_{split})) \Big/ \left[(K-1) \cdot p_{merge} \right] \tag{17}$$

5.3 Data Driven Proposal

In order to increase the efficiency of moves, proposals are all driven by the observation data. Let $p_{asso}(z_{t1}^i, z_{t2}^j)$ be an association likelihood between two neighboring observation node z_{t1}^i and z_{t2}^j, we define $p_{asso}(z_{t1}^i, z_{t2}^j)$ as:

$$p_{asso}(z_{t1}^i, z_{t2}^j) = \exp\left\{-\eta_1 \delta(L(z_{t1}^i) \neq L(z_{t2}^j)) - \eta_2 U_m(z_{t1}^i \mid z_{t2}^j)\right\} \quad (18)$$

where δ is the Dirac function, and U_m is defined as in (10). Denote neighboring relation between tracks as $\tau_m \sim \tau_n$, Then $p_d(z_i)$, $p(\tau(t_{split}))$ and $p_{merge}(\tau_i, \tau_j)$ in proposals take the formulation as follows:

$$p_d(z_i) = p_{asso}(z_i, z_c) \Big/ \sum_{z_j \in N^d(z_c)} p_{asso}(z_j, z_c) \quad (19)$$

$$p(\tau(t_{split})) = (1 - p_{asso}(\tau(t_{split}+1), \tau(t_{split})) \Big/ \sum_{i=2}^{|\tau|-1} (1 - p_{asso}(\tau(t_{i+1}), \tau(t_i))) \quad (20)$$

$$p_{merge}(\tau_i, \tau_j) = p_{asso}(\tau_i(t_{|\tau_i|}), \tau_j(t_1)) \Big/ \sum_{\tau_m \sim \tau_n} p_{asso}(\tau_m(t_{|\tau_m|}), \tau_n(t_1)) \quad (21)$$

6 Results

We test our algorithm on World Cup 2006 MPEG2 videos (image size: 720x576, 25 fps). First, our detection algorithm (without tracking) is tested on the France vs. Spain and Brazil vs. Japan videos. We randomly select hundreds of frames from each video (50-100 frames interval between adjacent selected frames) and manually labeled the ground truth. The performance of detection is shown in Table 1.

Video	Correct	False	Missing	Precision(%)	Recall(%)	F-score(%)
FRA vs. SPA	1133	145	96	88.65	92.19	90.39
BRA vs. JAP	5927	489	746	92.38	88.82	90.57

Table 1: Detection performance

We also test our labeling performance on the ground truth. Only those correctly detected players are used for evaluation. The confusion matrix of labeling is shown in Table 2, with GT stands for *ground truth*, INF stands for *inferred*. R denotes Referee, TA denotes Team A and TB denotes Team B. All the percentages are row-normalized.

GT \ INF	R (%)	TA(%)	TB(%)
R	82.93	17.07	0.00
TA	0.38	98.30	1.32
TB	0.00	2.31	97.69

(a) FRA vs. SPA

GT \ INF	R (%)	TA(%)	TB(%)
R	97.12	2.88	0.00
TA	1.23	97.75	1.02
TB	3.18	1.23	95.59

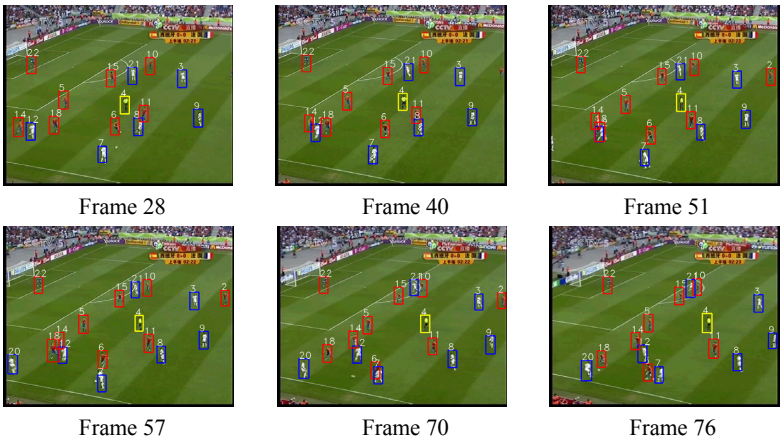
(b) BRA vs. JAP

Table 2: Confusion matrix of labeling with percentages row-normalized

We manually labeled a global view clip consisting of 100 consecutive frames from France vs. Spain video and test our tracking algorithm. The input observation set of these frames is shown in Figure 6(b). Some tracking results are shown in Figure 6(a). Automatic tracked players are enclosed by rectangles. The red rectangles denote “Team A”, blue denote “Team B” and yellow denote “Referee”. The ID of player is shown on

the top of each rectangle. It can be noticed that occlusion occurs between ID12, ID14 and ID18 from frame 40 to frame 70. Our algorithm correctly tracks and classifies them even if ID18 is totally occluded by ID12 in frame 51. Occlusion also occurs between ID8 and ID11 in frame 40, and between ID6 and ID7 in frame 70, which are all correctly handled by the algorithm.

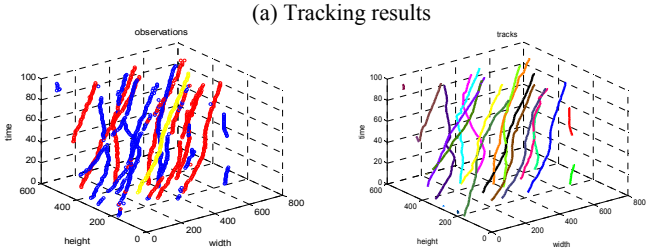
Figure 6(c) shows trajectories generated by tracking algorithm, with totally 24 players entering/leaving the scene. Our tracking algorithm can reach a precision of 99.32% and recall of 94.43% on the ground truth. Most miss detection is due to failure of boosting detection in several consecutive frames caused by video blur. It also can be noticed that the performance of labeling is significantly improved after tracking.



Frame 28 Frame 40 Frame 51

Frame 57 Frame 70 Frame 76

(a) Tracking results



(b) Observations (c) Associated tracks

Figure 6: Tracking results and generated trajectories

7 Conclusion

In this paper, we proposed an automatic player detection, unsupervised labeling and efficient player tracking method applied for broadcast soccer videos. The detection module combines background modeling and boosting detection. Labeling is achieved through unsupervised player appearance learning. MCMC data association is applied for tracking players. The result can be utilized for team tactics and player activity analysis, high-light detection, etc. The method can also be applied to other applications such as surveillance and vision-based human-computer interaction. While most of players can

be detected and tracked by our method, some cases such as long occlusions, serious video blur, abrupt camera motion and player tangle may still lead to failure. We plan to design more efficient MCMC proposals, and improve the labeling and tracking performance by playfield registration and trajectories inference in future works.

References

[1] M. Isard and J. MacCormick, "BraMBLe: A Bayesian Multiple-Blob Tracker," In Proc. IEEE Int'l Conf. on Computer Vision (ICCV), 2001.

[2] K. Okuma, A. Taleghani, N. de Freitas, J. J. Little, and D. G. Lowe, "A Boosted Particle Filter: Multitarget Detection and Tracking," In Proc. European Conf. on Computer Vision (ECCV), 2004

[3] Y. Bar-Shalom and T. Fortmann, Tracking and Data Association. San Diego, CA: Academic Press, 1988.

[4] D. Reid, "An Algorithm For Tracking Multiple Targets," IEEE Transaction on Automatic Control, vol. 24, no. 6, pp. 843–854, December 1979.

[5] S. Oh, S. Russell, and S. Sastry, "Markov Chain Monte Carlo Data Association for Multiple-Target Tracking," University of California, Berkeley, Technical Report UCB//ERL M05/19, June 2005.

[6] Q. Yu, G. Medioni, "Map-Enhanced Detection and Tracking from a Moving Platform with Local and Global Data Association," IEEE Workshop on Motion and Video Computing (WMVC'07), 2007.

[7] A.G.A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, Wensheng Hu, "Multi-Object Tracking Through Simultaneous Long Occlusions and Split-Merge Conditions," IEEE Int'l Conf. on Computer Vision and Pattern Recognition (CVPR), 2006.

[8] C. Stauffer, "Estimating Track Sources and Sinks", In Proc. IEEE Workshop on Event Mining in Video, 2003.

[9] R. Kaucic, A. G. A. Perera, G. Brooksby, J. Kaufhold, A. Hoogs, "A Unified Framework for Tracking Through Occlusions and Across Sensor Gaps", In Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition (CVPR), 2005.

[10] J. Sullivan and S. Carlsson, "Tracking and Labeling of Interacting Multiple Targets," In Proc. European Conf. on Computer Vision (ECCV), 2006.

[11] P. Nillius, J. Sullivan and S. Carlsson, "Multi-Target Tracking – Linking Identities using Bayesian Network Inference," In Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition (CVPR), 2006.

[12] J. Li, T. Wang, W. Hu, Y. Zhang, "Soccer Highlight Detection using Two-Dependent Bayesian Network," In Proc. of IEEE International Conf. on Multimedia & Expo (ICME), 2006.

[13] P. Viola, M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," In Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition (CVPR), 2001.

[14] J. Shotton, J. Winn, C. Rother, A. Criminisi, "TextonBoost: Joint Appearance, Shape and Context Modeling for Multi-Class Object Recognition and Segmentation," In Proc. European Conf. on Computer Vision (ECCV), 2006.

[15] J.A. Bilmes, "A Gentle Tutorial of the EM Algorithm and Its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models," TR-97-021, Berkeley, April 1998.

[16] J.S. Liu, Monte Carlo Strategies in Scientific Computing, Springer, 2001.

Feature-Driven Direct Non-Rigid Image Registration

V. Gay-Bellile[1,2] A. Bartoli[1] P. Sayd[2]
[1] LASMEA, CNRS / UBP, Clermont-Ferrand, France
[2] LCEI, CEA LIST, Saclay, France
Vincent.Gay-Bellile@univ-bpclermont.fr

Abstract

The registration problem for images of a deforming surface has been well studied. Parametric warps, for instance Free-Form Deformations and Radial Basis Functions such as Thin-Plate Spline, are often estimated using additive Gauss-Newton-like algorithms. The recently proposed compositional framework has been shown to be more efficient, but can not be directly applied to such non-groupwise warps.

We bring two contributions. First, we propose a Feature-Driven approach making possible the use of compositional algorithms for most parametric warps such as those mentioned above. This is demonstrated by an Inverse Compositional algorithm for Thin-Plate Spline warps. Second, we propose a piecewise linear learning approach to the local registration problem. Experimental results show that the basin of convergence is enlarged, computational cost reduced and alignment accuracy improved compared to previous methods.

1 Introduction

Registering images of a deforming surface is important for tasks such as video augmentation by texture editing, non-rigid Structure-from-Motion and deformation capture. This is a difficult problem since the appearance of imaged surfaces varies due to several phenomena such as camera pose, surface deformation, lighting and motion blur. Recovering a 3D surface, its deformations and the camera pose from a monocular video sequence is intrinsically ill-posed. While prior information can be used to disambiguate the problem, see *e.g.* [8, 13], it is common to avoid a full 3D model by using image-based deformation models, *e.g.* [2, 4, 6, 10]. TPS (Thin-Plate Splines) warps are one possible deformation model proposed in a landmark paper by Bookstein [4], that has been shown to effectively model a wide variety of image deformations in different contexts, including medical images. Recent work shows that TPS warps can be estimated with direct methods, *i.e.* by minimizing the intensity discrepancy between registered images [2, 10]. The Gauss-Newton algorithm with additive update of the parameters is usually used for conducting the minimization. Its main drawback is that the Hessian matrix must be recomputed and inverted at each iteration. More efficient solutions have been proposed by Baker *et al.* [1] based on a compositional update of the parameters, and lead to a constant Hessian matrix.

Most non-rigid warps do not from groups, preventing the use of compositional algorithms since they require to compose and possibly invert the warps. Despite several attempts to relax the groupwise assumption by various approximations [8, 12, 14], there is no simple solution in the literature.

This paper brings two main contributions. The first one is the *Feature-Driven* registration concept, allowing to devise compositional algorithms by relaxing the groupwise requirement for most non-rigid warps such as Radial Basis Functions (with *e.g.* the Thin-Plate Spline [4], a multiquadric [11] or the Wendland kernel function [7]). The main idea is to control the warp by a set of *driving features* (*e.g.* for Radial Basis Function warps, the target centers are used as driving features), and to act on these features directly for operations such as warp reversion and threading, approximating inversion and composition which are not guaranteed to exist or can not be easily computed. For instance, we extend the Inverse Compositional algorithm to TPS warps. The second contribution is an improvement of Cootes' linear learning approach [5] to local registration. Previous work assumes a linear relationship between the intensity discrepancy and the local parameter update [5, 9]. This assumption induces several practical problems, such as low alignment accuracy. We use a piecewise linear relationship for which a statistical map selection criterion is proposed. We experimentally show that it performs much better than most previous methods in terms of alignment accuracy, computational cost and enlarges the convergence basin. The combination of the Feature-Driven framework with learning-based local registration outperforms other algorithms for most experimental setups.

Notation. The images to be registered are written $\mathscr{I}_i$ with $i = 1, \ldots, n$. The template, *e.g.* the region of interest in the first image, is denoted $\mathscr{I}_0$. The parameterized warp is written $\mathscr{W}$. It depends on a parameter vector $\mathbf{u}_i$ for image $\mathscr{I}_i$ and maps a point $\mathbf{q}$ in the template to the corresponding point $\mathbf{q}_i$ in the i-th image: $\mathbf{q}_i = \mathscr{W}(\mathbf{q}; \mathbf{u}_i)$. We write $\mathscr{R}$ the set of pixels of interest and $\text{vect}_{\mathscr{R}}(\mathscr{M})$ the operator that vectorizes the elements of $\mathscr{M}$ indicated in $\mathscr{R}$.

2 Previous Work

The registration of images of deformable surfaces has received a growing attention over the past decade. Direct registration consists in minimizing the pixel value discrepancy. Registration of an image sequence is posed as a set of nonlinear optimization problems, each of which estimating $\mathbf{u}_{i+1}$ using the registration $\mathbf{u}_i$ of the previous frame as an initial solution. The discrepancy function $\mathscr{C}$ is usually chosen as the two-norm of the difference $\mathscr{D}$ between the template and the current one, warped towards the template, *i.e.* $\mathscr{D}(\mathbf{q}) = \mathscr{I}_0(\mathbf{q}) - \mathscr{I}_{i+1}(\mathscr{W}(\mathbf{q}; \mathbf{u}_{i+1}))$, giving: $\mathscr{C}(\mathbf{u}_{i+1}) = \sum_{\mathbf{q} \in \mathscr{R}} \|\mathscr{D}(\mathbf{q})\|^2$. Other choices are possible, such as Mutual Information.

Using an additive update of the parameter vector, *i.e.* $\mathbf{u}_{i+1} \leftarrow \mathbf{u}_{i+1} + \delta$, Gauss-Newton can be used in a straightforward manner for minimizing $\mathscr{C}$ or in conjunction with complexity tuning schemes as in [2, 10] for TPS warps. A second order approximation of $\mathscr{C}$, theoretically better than the Gauss-Newton one, is proposed in [3]. The major drawback of these methods is that the image gradient vector for each pixel in $\mathscr{R}$ must be recomputed at each iteration. This is the most expensive step of the process.

A major improvement was proposed by Baker and Matthews [1] through the Inverse Compositional algorithm. The key idea is to replace the additive update by the composition of the current warp $\mathscr{W}_{i+1}$ with the inverse $\tilde{\mathscr{W}}$ of the incremental warp: $\mathscr{W}_{i+1} \leftarrow \mathscr{W}_{i+1} \circ \tilde{\mathscr{W}}$. This leads to a constant Jacobian matrix and a constant Hessian matrix whose

inverse is thus pre-computed. This requires that the warp forms a group. In order to extend the approach to more involved models, several attempts have been made to relax the groupwise requirement. They are reviewed in §3.3. Previous work on learning-based registration are reviewed in §4.3.

3 Feature-Driven Registration

3.1 Feature-Driven Parametrization

The backbone of our approach is to represent the warp by a set of features in the current image, that we call *driving features*. These features have a fixed position in the template, depending on the type of warp that is being used. For RBF warps such as TPS warps, they can be placed anywhere, while for Free-Form Deformations, they must lie on a grid.

Henceforth, we assume that $\mathbf{u}_i$ contains the coordinates of the driving features in $\mathscr{I}_i$. In our implementation, we use TPS warps, whose Feature-Driven parameterization based on points is described in §A. The target centers of the TPS are used as driving features. The Feature-Driven concept and the registration algorithm we propose are generic in the sense that they are independent of the type of warp that is being used.

In this context, the warp is essentially seen as an interpolant between the driving features. There is obviously an infinite number of such warps. The best one depends on the nature of the observed surface. Loosely speaking, matching the driving features between two images is equivalent to defining a warp since the warp can be used to transfer the driving features from one image to the other, while conversely, the warp can be computed from the driving features.

The Feature-Driven framework has two main advantages. First, it often is better balanced to tune feature positions, expressed in pixels, than coefficient vectors that may be difficult to interpret, as for TPS warps. Second, it allows one to use the efficient compositional framework in a straightforward manner. Indeed, warp composition and inversion can not be directly done for non-groupwise warps. Representing image deformations by TPS warps or Free-Form Deformations is empirical. We propose empirical means for abstracting warp composition and inversion through their driving features, called threading and reversion respectively.

3.1.1 Threading Warps

Given two sets of driving features, $\mathbf{v}$ and $\mathbf{v}'$, we are looking for a third set $\mathbf{v}''$ defined such that threading the warps induced by $\mathbf{v}$ and $\mathbf{v}'$ results in the warp induced by $\mathbf{v}''$, as shown on figure 1(a). We propose a simple and computationally cheap way to do it, as opposed to previous work. This is possible thanks to the Feature-Driven parametrization. The idea is to apply the $\mathbf{v}'$ induced warp to the features in $\mathbf{v}$: the resulting set of features is $\mathbf{v}''$. This is written: $\mathbf{v}'' = \mathscr{W}(\mathbf{v};\mathbf{v}')$, where $\mathscr{W}$ is meant to be applied to each feature in $\mathbf{v}$. In the case of TPS warps, for which we use points as driving features, threading two warps is straightforward. It is also straightforward for all other kinds of RBF warps and for FFD warps.

3.1.2 Reverting Warps

Given a set $\mathbf{v}$ of driving features, we are looking for a set $\mathbf{v}'$, defined such that the warp induced by $\mathbf{v}'$ is the reversion of the one induced by $\mathbf{v}$, as illustrated on figure 1(b). As for

the threading, the Feature-Driven framework makes a very simple solution possible. The idea is that applying the $\mathbf{v}'$ induced warp to $\mathbf{v}$ gives $\mathbf{u}_0$, *i.e.* the fixed driving features in the template. This is written: $\mathscr{W}(\mathbf{v};\mathbf{v}') = \mathbf{u}_0$. This is straightforward for TPS warps. This amounts to solving an exactly determined linear system, the size of which is the number of driving features. For some classes of warps, $\mathscr{W}(\mathbf{v}';\mathbf{v}) = \mathbf{u}_0$ may be more practical to solve for $\mathbf{v}'$. Note that for all other kinds of RBF warps and FFD warps $\mathscr{W}(\mathbf{v};\mathbf{v}') = \mathbf{u}_0$ is to be preferred as well.

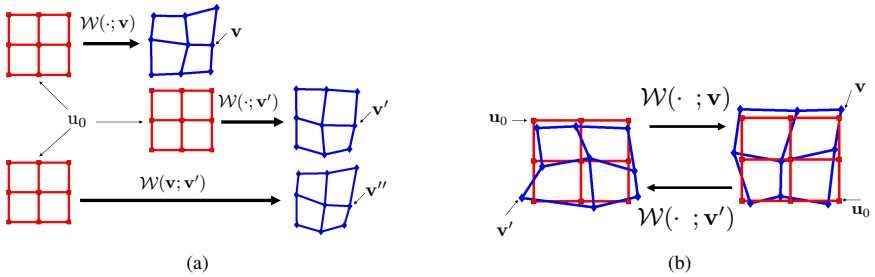


Figure 1: (a) The Feature-Driven warp threading process: $\mathbf{v}''$ is defined by $\mathbf{v}'' = \mathscr{W}(\mathbf{v};\mathbf{v}')$. (b) The Feature-Driven warp reversion process: $\mathbf{v}'$ is defined such that $\mathscr{W}(\mathbf{v};\mathbf{v}') = \mathbf{u}_0$.

3.2 Compositional Feature-Driven Alignment

Benefiting from the Feature-Driven parameterization properties, we extend the compositional algorithms to non-groupwise warps. The following three steps, illustrated on figure 2 are repeated until convergence:

- **Step 1: Warping.** The driving features $\mathbf{u}_{i+1}$ are used to warp the input image $\mathscr{I}_{i+1}$, thereby globally aligning it to the template: $\mathscr{I}_W(\mathbf{q}) = \mathscr{I}_{i+1}(\mathscr{W}(\mathbf{q};\mathbf{u}_{i+1}))$.
- **Step 2: Local alignment.** The driving features $\mathbf{u}$ are estimated in the warped image. This is described in §4. Note that for the Inverse Compositional algorithm, warp reversion is done at this step.
- **Step 3: Updating.** The current driving features $\mathbf{u}_{i+1}$ and those in the warped image $\mathbf{u}$ are combined by threading the warps to update the driving features $\mathbf{u}_{i+1}$ in the current image: $\mathbf{u}_{i+1} \leftarrow \mathscr{W}(\mathbf{u};\mathbf{u}_{i+1})$.

Note that previous work [8, 12, 14] requires a preliminary step before applying the update rule, as reviewed in the next section. In comparison, the Feature-Driven framework makes it naturally included into the third step.

Illumination changes are handled by normalizing the pixel value of the template and those of the warped image at each iteration.

3.3 Relation to Previous Work

Alternative approaches for non-groupwise warp composition as proposed in [8, 12, 14] consist in finding the best approximating warp for the pixels of interest in $\mathscr{R}$: $\mathbf{u}_{i+1} = \arg\min_{\mathbf{u}_{i+1}} \sum_{\mathbf{q}\in\mathscr{R}} \|\mathscr{W}_i(\tilde{\mathscr{W}}(\mathbf{q})) - \mathscr{W}_{i+1}(\mathbf{q})\|^2$. In [8, 12, 14] the warp is induced by a triangular mesh whose deformations are guided by a parameter vector. This minimization

problem is usually solved in two steps. First the vertices in the current image are computed using the assumption of local rigidity. They usually are not in accordance with a model instance in *e.g.* the case of 3D Morphable Models [8, 14]. Second, the parameter update is recovered by minimizing a prediction error, *i.e.* the distance between the updated vertices and those induced by the parameters. This last step may be time consuming since nonlinear optimization is required. Warp inversion is approximated with first order Taylor expansion in [12], while [14] draws on triangular meshes to avoid linearization. By comparison, our methods revert and thread warps in closed-form: they do not require optimization.

4 Local Registration

The efficiency of compositional algorithms depends on the local alignment step. For instance, the Inverse Compositional algorithm is efficient with the Gauss-Newton approximation for local alignment since this combination makes invariant the Hessian matrix. We show that learning-based local alignment fits in the Forward Compositional framework in a similar manner. Below, we propose an efficient learning-based alignment procedure. This is inspired by previous work modeling the relationship between the local increment δ and the intensity discrepancy $\mathbf{D}$ with an interaction matrix. Using a single interaction matrix has several drawbacks, as reviewed in §4.3. We propose to learn a series $\mathscr{F}_1 \dots \mathscr{F}_\kappa$ of interaction matrices, each of them covering a different range of displacement magnitudes. This forms a piecewise linear approximation to the true relationship. Each matrix thus defines a map. A statistical map selection procedure is learned in order to select the most appropriate matrix $\mathscr{F}_s$ given $\mathbf{D}$. The update vector is then given by: $\delta = \mathscr{F}_s \mathbf{D}$. Details are given below.

4.1 Learning an Interaction Matrix

An interaction matrix $\mathscr{F}$ is learned by generating artificially perturbed versions of the template $\mathscr{A}^j$.

Generating training data. The driving features in the template are disturbed from their rest position with randomly chosen direction and magnitude: $\mathbf{u}^j \leftarrow \mathbf{u}_0 + \delta\mathbf{u}^j$. The latter is clamped between a lower and an upper bound, determining the area of validity of the interaction matrix. Our Feature-Driven warp reversion process is used to warp the template.

Learning. The residual vector is computed from the pixels of interest in $\mathscr{R}$: $\mathbf{D}^j = \text{vect}_{\mathscr{R}}\left(\mathscr{I}_0 - \mathscr{A}^j\right)$. The training data are gathered in matrices $\mathscr{U} = \left(\delta\mathbf{u}^1|...|\delta\mathbf{u}^m\right)$ and $\mathscr{L} = \left(\mathbf{D}^1|...|\mathbf{D}^m\right)$. The interaction matrix $\mathscr{F}$ is computed by minimizing a Least Squares error in the image space, expressed in pixel value unit: $\mathscr{F} = \left(\mathscr{L}\mathscr{U}^\mathsf{T}(\mathscr{U}\mathscr{U}^\mathsf{T})^{-1}\right)^\dagger$. This is one of the two possibilities for learning the interaction matrix. The other possibility is dual. It minimizes an error in the parameter space, *i.e.* expressed in pixels. Experimental results show that the former gives better results, being in particular much more robust to noise.

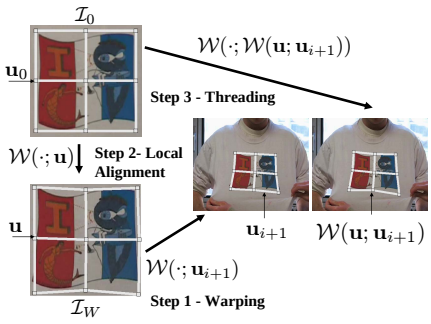


Figure 2: Compositional Feature-Driven Alignment .

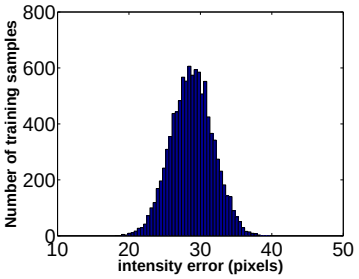


Figure 3: Distribution example of the intensity error magnitude for a perturbation interval: $[7\ldots13]$ pixels.

4.2 Statistical Map Selection

One issue with the piecewise linear model is to select the best interaction matrix at each iteration. Each of those indeed has a specific domain of validity in the displacement magnitude which unfortunately can not be determined prior to image alignment. We propose to learn a relationship between the intensity error magnitude and the best matrix to use. We express this relationship in terms of probabilities. The intensity error magnitude for a residual error vector $\mathbf{D}$ is defined as its RMS (Root Mean of Squares) $e(\mathbf{D})$. Note that $e^2(\mathbf{D}) \propto \mathscr{C}(\mathbf{u}_{i+1})$. Figure 3 shows that $P(\mathscr{F}_k|e(\mathbf{D}))$ closely follows a Gaussian distribution. The mean and variance of the learned intensity error magnitude are computed for each interaction matrix. Finding the most appropriate interaction matrix given the current intensity error is simply achieved by solving: $s = \arg\max_t P(\mathscr{F}_t|e(\mathbf{D}))$.

4.3 Relation to Previous Work

Learning approaches in the literature often assume that the relationship between the error image and the update parameters is linear [5, 9]. The drawback of those methods is that if the interaction matrix covers a large domain of deformation magnitudes, alignment accuracy is spoiled. On the other hand if the matrix is learned for small deformations only, the converge basin is dramatically reduced. Our piecewise linear relationship solves these issues.

Interaction matrices are valid only locally around the template parameters. Compositional algorithms are thus required, as in [9] for homographic warps. However in [5] the assumption is made that the domain where the linear relationship is valid covers the whole set of registrations. They thus apply the single interaction matrix around the current parameters, avoiding the warping and the composition steps. This does not appear to be a valid choice in practice. Our Feature-Driven framework naturally extends this approach to non-groupwise warps.

5 Experimental Results

We compare four algorithms in terms of convergence frequency, accuracy and convergence rate. Two classical algorithms:

- FA-GN: the Forward Additive Gauss-Newton approach used by [2, 10].
- FA-ESM: the Efficient Second Order registration algorithm [3], adapted to TPS warps.
Two algorithms we propose:
- IC-GN: the Feature-Driven Inverse Compositional registration of §3 with Gauss-Newton as local registration engine.
- FC-Le: the Feature-Driven Forward Compositional registration of §3, with local registration achieved through learning as we propose in §4.

5.1 Simulated Data

In order to assess the algorithms in different controlled conditions, we synthesize images from a template. The driving features are placed on a 3×3 grid, randomly perturbed with magnitude r. We add Gaussian noise, with variance $\sigma\%$ of the maximum greylevel value, to the warped image. We vary each of these parameters independently, using the following default values: $r = 2$ pixels and $\sigma = 1\%$. The noise variance upper bound is 10%, which corresponds to very noisy images. Estimated warps are scored by the mean Euclidean distance between the driving features which generated the warped image, and the estimated ones. Convergence to the right solution is declared if this score is lower than one pixel. The results are means over 500 trials.

Convergence frequency. This is the percentage of convergence to the right solution. Results are shown on figures 4(a) and 4(b). FC-Le has the largest convergence basin closely followed by FA-ESM. On the other hand, FC-Le has the worst performance against noise. However, it always converges for noise amplitude below 8% and converges at 95% for 10% noise which is beyond typical practical values. IC-GN has the smallest convergence basin.

Accuracy. This is measured as the mean residual error over the trials for which an algorithm converged. Results are shown on figures 4(c) and 4(d). The four algorithms are equivalent against displacement magnitude. Concerning noise amplitude, IC-GN and FC-Le are equivalent while FA-ESM is slightly worse and FA-GN clearly worse. For example, at 6% noise, the alignment errors of IC-GN and FC-Le are around 0.2 pixels, FA-ESM at about 0.25 pixels and FA-GN alignment error at 0.35 pixels.

Convergence rate. This is defined by the number of iterations required to converge. Results are shown on figures 4(e) and 4(f). The convergence rate of FC-LE and FA-ESM are almost constant against both displacement and noise amplitudes. However FC-Le does better, with a convergence rate kept below 10. FA-GN and IC-GN are efficient for small displacements, *i.e.* below 5 pixels. The convergence rate increases dramatically beyond this value for both of them. FA-GN is also inefficient for noise amplitude over 4%. This is explained by the fact that the FA-GN Jacobian matrix depends mainly on the gradient of the current image, onto which the noise is added.

5.2 Real Data

The four above described algorithms have been compared on several videos. For two of them (*paper*, *tshirt*), we show results on table 1 and registration visualization samples on figure 5. We measure the average and maximum intensity RMS along the video, computed on the pixels of interest and expressed in pixel value unit, the total number of iterations and the computational time, expressed in seconds. All algorithms have been implemented in

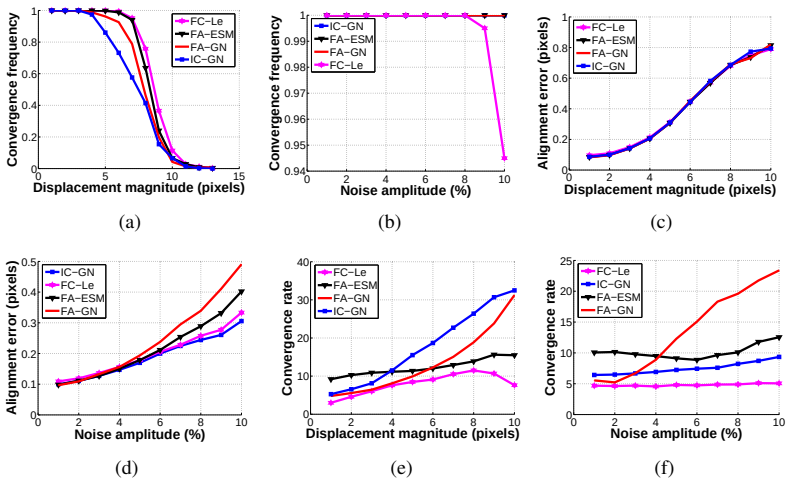


Figure 4: Comparison of the four algorithms in terms of convergence frequency (a) against displacement magnitude and (b) noise magnitude. Comparison of the four algorithms in terms of accuracy (c) against displacement magnitude and (d) noise magnitude. Comparison of the four algorithms in terms of convergence rate (e) against displacement magnitude and (f) noise magnitude.

Matlab. In practice, a great number of driving features with some amount of regularization are used, making the TPS both flexible and well-constrained. In order to illustrate the registration, we define a mesh on the template and transfer it to all other frames. Note that these meshes are different from the estimated driving features. The alignment differences between the four algorithms are visually undistinguishable, when they converge.

Summary. FA-GN is an accurate algorithm. It is however inefficient, especially for important displacements. FA-ESM has almost similar performances compared to FA-GN while being slightly more efficient. The Feature-Driven parametrization yields, via the proposed IC-GN and FC-Le algorithms, fast non rigid registration with TPS warps. While IC-GN looses effectiveness for high displacements, FC-Le has the best behavior. In fact, it is similar to FA-GN for accuracy while being 5 times faster on average and is equivalent or better than IC-GN and FA-ESM in terms of alignment accuracy, computational cost and has a larger convergence basin.

	Mean/max RMS		Iteration #		Total/mean time	
	tshirt	*paper*	*tshirt*	*paper*	*tshirt*	*paper*
FA-GN	8.7/13.6	**8.98/17.57**	9057	2422	2083/5.2	702/2.0
FA-ESM	9.2/14.7	10.22/20.49	3658	2473	877/2.2	708/2.0
IC-GN	9.7/15.8	Diverges	6231	Diverges	436/1.1	Diverges
FC-Le	**6.66/12.87**	9.44/19.4	**3309**	**1330**	**380/0.95**	**176/0.5**

Table 1: Results for the *tshirt* and *paper* videos. Bold indicates best performances.

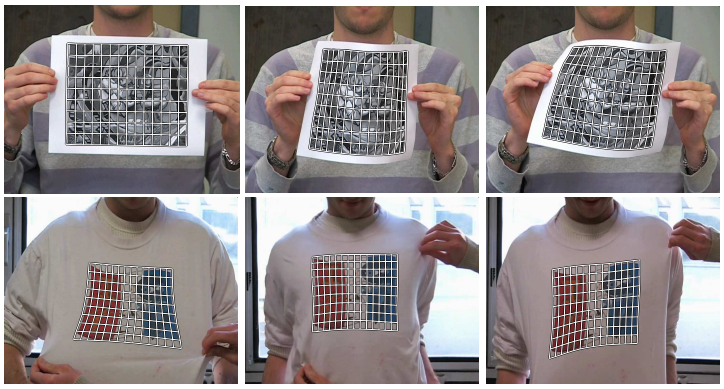


Figure 5: Registration results for FC-Le algorithm. Top: the *paper* video. Bottom: the *tshirt* video.

6 Conclusions

We addressed two important issues for the problem of non-rigid registration. First, we proposed the Feature-Driven framework, relaxing the groupwise requirement for using efficient compositional algorithms. Second, we proposed a statistically motivated piecewise linear local registration engine. Combining these two techniques results in an algorithm outperforming the other ones in terms of alignment accuracy, computational cost and having a larger convergence basin. Real-time surface registration is foreseen with this algorithm. We intend to extend the Feature-Driven approach to more complicated models, *e.g.* 3D Morphable Models [14]. This implies occlusion reasoning, that we intend to do through multiple overlapping patch registration and combination.

References

[1] S. Baker and I. Matthews. Lucas-Kanade 20 years on: A unifying framework. *IJCV*, 2004.

[2] A. Bartoli and A. Zisserman. Direct estimation of non-rigid registrations. In *BMVC*, 2004.

[3] S. Benhimane and E. Malis. Real-time image-based tracking of planes using efficient second-order minimization. In *IROS*, 2004.

[4] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. IEEE *PAMI*, 1989.

[5] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *ECCV*, 1998.

[6] T. F. Cootes, S. Marsland, C. J. Twining, K. Smith, and C. J. Taylor. Groupwise diffeomorphic non-rigid registration for automatic model building. In *ECCV*, 2004.

[7] Mike Fornefett, Karl Rohr, and H. Siegfried Stiehl. Elastic registration of medical images using radial basis functions with compact support. In *CVPR*, 1999.

[8] V. Gay-Bellile, M. Perriollat, A. Bartoli, and P. Sayd. Image registration by combining thin-plate splines with a 3D morphable model. In *ICIP*, 2006.

[9] F. Jurie and M. Dhome. Hyperplane approximation for template matching. IEEE *PAMI*, 2002.

[10] J. Lim and M.-H. Yang. A direct method for non-rigid motion with thin-plate spline. In *CVPR*, 2005.

[11] J. A. Little, D. L. G. Hill, and D. J. Hawkes. Deformations incorporating rigid structures. *CVIU*, 1997.

[12] I. Matthews and S. Baker. Active appearance models revisited. *IJCV*, 2004.

[13] J. Pilet, V. Lepetit, and P. Fua. Real-time non-rigid surface detection. In *CVPR*, 2005.

[14] S.Romdhani and T.Vetter. Efficient, robust and accurate fitting of a 3D morphable model. In *ICCV*, 2003

A Feature-Driven Thin-Plate Spline Warps

This parameterization is convenient for centers that remain still in the template. The TPS is an $\mathbb{R}^2 \to \mathbb{R}$ function driven by assigning target values α_k to l 2D centres $\mathbf{c}_k$ and enforcing several conditions: the TPS is the Radial Basis Function (RBF) that minimizes the integral bending energy. It is usually parameterized by an $l+3$ coefficient vector $\mathbf{h}_{\alpha,\lambda}^{\mathsf{T}} = (\ \mathbf{w}^{\mathsf{T}}\ \ \mathbf{a}^{\mathsf{T}})$ computed from the target vector α and a regularization parameter $\lambda \in \mathbb{R}^+$. The coefficients in $\mathbf{w}$ must satisfie $\mathsf{P}^{\mathsf{T}}\mathbf{w} = \mathbf{0}$, where the k-th row of P is $\left(\mathbf{c}_k^{\mathsf{T}}\ 1\right)$. These three 'side-conditions' ensure that the TPS has square integrable second derivatives. The TPS is defined by:

$$\omega(\mathbf{q}, \mathbf{h}_{\alpha,\lambda}) = \ell_{\mathbf{q}}^{\mathsf{T}} \mathbf{h}_{\alpha,\lambda}, \tag{1}$$

with $\ell_{\mathbf{q}}^{\mathsf{T}} = (\rho(d^2(\mathbf{q},\mathbf{c}_1)) \ \cdots\ \rho(d^2(\mathbf{q},\mathbf{c}_l))\ \mathbf{q}^{\mathsf{T}}\ 1)$. Combining the equations obtained for all the l centres $\mathbf{c}_r$ with target values α_r in a single matrix equation gives:

$$\mathsf{K}_\lambda \mathbf{w} + \mathsf{P}\mathbf{a} = \alpha,\ K_{r,k} = \begin{cases} \lambda & r = k \\ \rho(d^2(\mathbf{c}_r, \mathbf{c}_k)) & r \neq k. \end{cases} \tag{2}$$

Adding $\lambda \mathrm{I}$ acts as a regularizer. Solving for $\mathbf{h}_{\alpha,\lambda}$ using the above equation and the side-conditions is the classical linear method for estimating the TPS coefficients due to Bookstein [4]. The coefficient vector $\mathbf{h}_{\alpha,\lambda}$ is a nonlinear function of the regularization parameter λ and a linear function of the target vector α.

We write $\mathbf{h}_{\alpha,\lambda} = \mathscr{E}_\lambda \alpha$, *i.e.* as a linear 'back-projection' of the target vector α. Matrix $\mathscr{E}_\lambda$ nonlinearly depends on λ. It is given from (2) as a function of K_λ and P by:

$$\mathscr{E}_\lambda = \begin{pmatrix} \mathsf{K}_\lambda^{-1}\left(\mathrm{I} - \mathsf{P}\left(\mathsf{P}^{\mathsf{T}}\mathsf{K}_\lambda^{-1}\mathsf{P}\right)^{-1}\mathsf{P}^{\mathsf{T}}\mathsf{K}_\lambda^{-1}\right) \\ \left(\mathsf{P}^{\mathsf{T}}\mathsf{K}_\lambda^{-1}\mathsf{P}\right)^{-1}\mathsf{P}^{\mathsf{T}}\mathsf{K}_\lambda^{-1} \end{pmatrix}.$$

This parameterization has the advantages to separate λ and α and introduces units[1]. The side-conditions are naturally enforced by this parameterization.

Incorporating this parameterization into the TPS (1) we obtain what we call the *feature-driven* parameterization for the TPS: $\tau(\mathbf{q};\alpha,\lambda) = \ell_{\mathbf{q}}^{\mathsf{T}} \mathscr{E}_\lambda \alpha$. Standard $\mathbb{R}^2 \to \mathbb{R}^2$ TPS-Warps are obtained by stacking two $\mathbb{R}^2 \to \mathbb{R}$ TPS sharing their centres and regularization parameter:

$$\mathscr{W}(\mathbf{q};\mathbf{u},\lambda) = (\tau(\mathbf{q};\alpha_x,\lambda)\ \tau(\mathbf{q};\alpha_y,\lambda))^{\mathsf{T}} = \begin{pmatrix}\alpha_x & \alpha_y\end{pmatrix}^{\mathsf{T}} \mathscr{E}_\lambda^{\mathsf{T}} \ell_{\mathbf{q}}, \tag{3}$$

with $\mathbf{u}^{\mathsf{T}} = \begin{pmatrix}\alpha_x^{\mathsf{T}}\ \alpha_y^{\mathsf{T}}\end{pmatrix}$. Notation $\mathscr{W}(\mathbf{q};\mathbf{u})$ is used for $\lambda = 0$.

[1] While $\mathbf{h}_{\alpha,\lambda}$ has no obvious unit, α in general has (*e.g.* pixels, meters).

Discovering Planes and Collapsing the State Space in Visual SLAM

Andrew P. Gee, Denis Chekhlov, Walterio Mayol and Andrew Calway

Department of Computer Science
University of Bristol, UK

{gee,chekhlov,wmayol,andrew}@cs.bris.ac.uk

Abstract

Recent advances in real-time visual SLAM have been based primarily on mapping isolated 3-D points. This presents difficulties when seeking to extend operation to wide areas, as the system state becomes large, requiring increasing computational effort. In this paper we present a novel approach to this problem in which planar structural components are embedded within the state to represent mapped points lying on a common plane. This collapses the state size, reducing computation and improving scalability, as well as giving a higher level scene description. Critically, the plane parameters are augmented into the SLAM state in a proper fashion, maintaining inherent uncertainties via a full covariance representation. Results for simulated data and for real-time operation demonstrate that the approach is effective.

1 Introduction

Concurrently estimating the pose of a moving camera and scene structure in real-time is appealing, both for the algorithmic challenges involved and the potential applications that can result. Recently, much progress has been made by adopting the principles of simultaneous localisation and mapping (SLAM) from robotics [19], where it was aimed at enabling autonomous systems to navigate around previously unknown surroundings. For a single camera, however, either attached to a person or handheld, and using only visual measurements, the problem is much more difficult, as the benefits of wheel odometry and control in robotics, for example, are no longer available.

Since the pioneering work of Davison [7], based on Kalman filtering and efficient feature matching, several visual SLAM systems have been proposed, addressing problems such as scalability and robustness. For example, Eade and Drummond [10] pose the problem within a factored sampling framework, using the FastSLAM algorithm to offer better theoretical scalability of map size, whilst Chekhlov et al. [5] use highly discriminative features with scale prediction to improve robustness of tracking and to handle periods of measurement loss. More recently, Williams et al. [20] combine an auxiliary RANSAC process with the Kalman filter to achieve relocalisation when tracking fails.

These systems are based on sparse maps, which are built by estimating the position of 3-D scene points and their associated uncertainties. This minimal representation is attractive since, in practice, it allows real-time operation whilst providing sufficient information to maintain accurate tracking. Recently, however, there has been a move towards

the use of more structure-rich maps, in order to gain higher level scene descriptions and minimise redundancy. This includes the estimation of local surface normals [14], 3-D line segments [18, 12] and edgelets [9]. However, these descriptions are still based on isolated 3-D features and there is no sense of how these features group together to form higher level structures. A move towards the latter has been developed by Castle et al. [4], where known planar objects are detected using appearance features and inserted into the map.

In this paper we describe a novel approach to building in structural grouping within visual SLAM without prior knowledge of objects. This is achieved by augmenting the state with parameters representing higher level structures and then 'folding in' subsets of 3-D points which are consistent with the higher level parameterisation. This collapses the state space, reducing computational demands and increasing scalability, as well as giving a higher level scene description. Crucially, this is done in a manner which maintains the inherent uncertainties built up during SLAM operation by adopting a full covariance representation, ensuring consistency of the tracking and the map estimate. We demonstrate the approach by using it to discover 3-D planes in a scene. Plane detection has been previously investigated in the context of offline structure from motion [3] and is a natural choice to enhance visual maps given that most man-made environments contain large well defined planes. In Augmented Reality applications, for instance, planes are useful for augmenting the scene with virtual objects that are relative to a planar surface [13].

The paper is organised as follows. In the next section we briefly summarise the basic visual SLAM system. Details of state space augmentation and collapsing are then given in Section 3, followed by details of how this can be used for discovering higher level structure such as planar surfaces. Results are then presented for simulated data and real-time operation within an office environment.

2 Visual SLAM Using Kalman Filtering

We assume a calibrated camera moving with agile motion whilst viewing a static scene. The aim is to estimate the camera pose, whilst simultaneously estimating the 3-D parameters of structural features, such as points or surfaces. We use a Kalman filter framework, similar to that in [7, 18, 5]. The system state $\mathbf{x} = [\mathbf{v}, \mathbf{m}]^T$ has two partitions, one for the camera pose $\mathbf{v} = [\mathbf{q}, \mathbf{t}]^T$ and one for the structural features, $\mathbf{m} = [\mathbf{m}_1, \mathbf{m}_2, \ldots]^T$. The camera rotation is represented by the quaternion $\mathbf{q}$ and $\mathbf{t}$ is its 3-D position vector, both defined w.r.t a world coordinate system. We adopt a generalised framework for the structural partition to allow multiple feature types to be mapped. Each $\mathbf{m}_i$ denotes a parameterisation of a feature, be it a position vector for a 3-D point or the position and orientation of a planar surface. In general, we expect the number and the types of features to vary as the camera explores the environment, with new features being added and old ones being removed so as to minimise computation and maintain stability.

The Kalman filter requires a *process model* and an *observation model*, encoding the assumed evolution of the state between time steps and the relationship between the state and the filter measurements, respectively [2]. We assume a constant position model for the camera motion, i.e. a random walk, and since we are dealing with a static scene this gives the process model

$$\mathbf{x}^{new} = \mathbf{f}(\mathbf{x}, \mathbf{w}) = [\Delta\mathbf{q}(\mathbf{w}_\omega) \otimes \mathbf{q}, \mathbf{t} + \mathbf{w}_\tau, \mathbf{m}]^T \quad (1)$$

where $\mathbf{w} = [\mathbf{w}_\omega, \mathbf{w}_\tau]^T$ is a 6-D noise vector, assumed to be from $\mathbf{N}(0, \mathbf{Q})$, $\Delta\mathbf{q}(\mathbf{w}_\omega)$ is the incremental quaternion corresponding to the Euler angles defined by $\mathbf{w}_\omega$, and $\otimes$ denotes quaternion multiplication. Note that the map parameters are predicted to remain unchanged and that the non-additive noise component in the quaternion part is necessary to give an unbiased distribution in rotation space. At each time step, we collect a set of measurements, $\mathbf{z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots]^T$, where in general each $\mathbf{z}_i$ will be related to the state via a separate measurement function, i.e. $\mathbf{z}_i = \mathbf{h}_i(\mathbf{x}, \mathbf{e})$, where $\mathbf{e}$ is a multivariate noise vector from $\mathbf{N}(0, \mathbf{R})$. For example, when mapping point features, the measurements are assumed to be corrupted 2-D positions of projected points and for the *i*th measurement

$$\mathbf{h}_i(\mathbf{x}, \mathbf{e}) = \Pi(\mathbf{y}(\mathbf{v}, \mathbf{m}_j)) + \mathbf{e}_i \tag{2}$$

where $\mathbf{m}_j$ is the 3-D position vector of the point associated with the measurement $\mathbf{z}_i$, $\mathbf{y}(\mathbf{v}, \mathbf{m}_j)$ denotes this position vector in the camera coordinate system, Π denotes pin-hole projection for a calibrated camera and $\mathbf{e}_i$ is a 2-D noise vector from $\mathbf{N}(0, \mathbf{R}_i)$.

The filter gives state mean and covariance estimates based on the process and observation models. Both are non-linear and thus we use the extended Kalman filter, in which predictions and updates are derived from approximations based on the Jacobians of $\mathbf{f}$ and $\mathbf{h}_i$ [2]. The role of the covariances within the filter is critical. Proper maintenance ensures that updates are propagated amongst the state elements, particularly the structural components, which are correlated through their mutual dependence on the camera pose [19, 8]. This ensures consistency and stability of the filter. Propagation of the state covariance through the observation model also allows for better data association, constraining the search for image features. As discussed next, it is therefore crucial to ensure that covariances are properly maintained for successful filter operation.

3 EKF SLAM with Plane Discovery

We discover higher level structure within the SLAM framework in the following manner. We begin by mapping 3-D points, starting from known points on a calibration pattern [7]. As the camera moves away, new points are initialised and added to the map, allowing tracking to continue as the camera explores the scene. Points are added using the inverse depth representation [15] and augmented to the state in a manner which maintains full covariance with the rest of the map [1] (see below). For measurements, we use the fast salient point detector developed by Rosten and Drummond [17] to identify potential features and the SIFT-like descriptors with scale prediction developed by Chekhlov *et al.* [5] for repeatable matching across frames.

As SLAM proceeds, the system seeks to identify subsets of mapped points that potentially lie on a planar structure. This is done using an auxiliary RANSAC process and a subsequent principal component analysis amongst inliers then yields initial estimates for the plane parameters, allowing a new 'planar feature' to be augmented to the state. The 3-D inlier points are then transformed to 2-D planar points, representing their position within the plane, hence collapsing the state. Critically, both the augmentation and transformation include proper adjustment of the state covariance, ensuring that full cross correlation is maintained amongst the existing and new state parameters. As SLAM continues, new 3-D planes are initialised as appropriate and the system attempts to collapse new 3-D points into the planes, hence continuing to reduce state size. Details of each component are given in the following sections.

3.1 Augmenting and Collapsing the State Space

It is useful to consider first the general procedure for augmenting and transforming the state space in a SLAM system [1, 6]. Assume that we have an existing state estimate $\hat{\mathbf{x}} = [\hat{\mathbf{v}}, \hat{\mathbf{m}}_1, \ldots, \hat{\mathbf{m}}_n]^T$, with n features in the map, and that we wish to augment the state with a new feature $\mathbf{m}_{n+1}$, based on an initialisation measurement $\mathbf{z}_o$. In general, the initial estimate for the feature will be derived from a combination of the measurement and the existing state, i.e. $\hat{\mathbf{m}}_{n+1} = \mathbf{s}(\hat{\mathbf{x}}, \mathbf{z}_o)$, and the augmented state covariance then becomes

$$\mathbf{P}^{new} = \mathbf{J} \begin{bmatrix} \mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_o \end{bmatrix} \mathbf{J}^T \qquad \mathbf{J} = \left[\begin{array}{c|c} \mathbf{I} & \mathbf{0} \\ \hline \nabla\mathbf{s}_{\mathbf{v}} \;\; \nabla\mathbf{s}_{\mathbf{m}_1} \;\; \ldots \;\; \nabla\mathbf{s}_{\mathbf{m}_n} & \nabla\mathbf{s}_{\mathbf{z}_o} \end{array}\right] \tag{3}$$

where $\mathbf{R}_o$ is the covariance of the measurement and $\nabla\mathbf{s}_{\mathbf{v}} = \partial\mathbf{s}/\partial\mathbf{v}$. This transformation of the covariance introduces the important correlations between the new feature and the existing features in the map. For example, in the case of augmentation with a new point feature, only the Jacobians $\nabla\mathbf{s}_{\mathbf{v}}$ and $\nabla\mathbf{s}_{\mathbf{z}_o}$ are non-zero and the new point feature is correlated with those already in the map through the camera pose. In a similar way, we can also transform an existing feature to a different representation, e.g. $\hat{\mathbf{m}}_i^{new} = \mathbf{r}(\hat{\mathbf{x}})$, and then the covariance update is given by

$$\mathbf{P}^{new} = \mathbf{J}\mathbf{P}\mathbf{J}^T \qquad \mathbf{J} = \left[\begin{array}{c|c|c} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \hline \nabla\mathbf{r}_{\mathbf{v}} \;\; \ldots \;\; \nabla\mathbf{r}_{\mathbf{m}_{i-1}} & \nabla\mathbf{r}_{\mathbf{m}_i} & \nabla\mathbf{r}_{\mathbf{m}_{i+1}} \;\; \ldots \;\; \nabla\mathbf{r}_{\mathbf{m}_n} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{I} \end{array}\right] \tag{4}$$

where any change in the dimensions of the state is reflected in the dimensions of the Jacobian $\mathbf{J}$. We can use both of the above relationships in order to introduce higher level features and collapse the state size. For example, consider a new feature $\mathbf{m}_{n+1}$ which represents a higher level structure and which imposes a constraint on a subset of existing features. Given an initial estimate of $\mathbf{m}_{n+1}$, we can introduce it into the system using (3) and the constraint can then be imposed by transforming the existing features into the new constrained representation using (4). This collapses the state by removing redundant constrained variables, whilst also maintaining the correlations between the new features, the camera pose and the remainder of the map.

3.2 Representing Planes

We demonstrate the process of collapsing the state space using the example of discovering and enforcing planar structure in a scene. The descriptors used to match and identify 3-D points perform best when they are initialised on planar, textured surfaces, so it is likely that many of the points in the map will end up lying on planes. We seek to infer the locations and orientations of these planes and introduce parameterisations of the planes as new features in the map. This then allows the 3-D points lying on the planes to be collapsed into 2-D planar points defined w.r.t the plane. We use a seven parameter state vector to define a plane, so that a new planar feature augmented to the state has the form $\mathbf{m}_{n+1} = [\mathbf{p}_o, \theta_1, \phi_1, \theta_2, \phi_2]^T$, where $\mathbf{p}_o$ is the plane origin and the orientation is defined by two basis vectors, $\mathbf{c}(\theta_1, \phi_1)$ and $\mathbf{c}(\theta_2, \phi_2)$, which lie on the plane, i.e.

$$\mathbf{c}(\theta_i, \phi_i) = [cos\phi_i \; sin\theta_i, \; -sin\phi_i, \; cos\phi_i \; cos\theta_i]^T \tag{5}$$

Note that the normal to the plane is then simply the cross product between the two basis vectors, i.e. $\mathbf{n} = \mathbf{c}(\theta_1, \phi_1) \times \mathbf{c}(\theta_2, \phi_2)$. A 3-D point in the map which lies on the plane and whose feature vector $\mathbf{m}_i$ defines its 3-D position vector, can then be transformed into a 2-D planar point using

$$\mathbf{m}_i^{new} = [(\mathbf{m}_i - \mathbf{p}_o) \cdot \mathbf{c}(\theta_1, \phi_1), \ (\mathbf{m}_i - \mathbf{p}_o) \cdot \mathbf{c}(\theta_2, \phi_2)]^T \tag{6}$$

where $\cdot$ denotes the dot product. Thus for l such points, this gives a state size of $7 + 2l$, compared with $3l$, giving a reduction in state size for $l > 7$.

3.3 Initialising Planes

We use the RANSAC algorithm [11] to search for planes amongst point features in the map, similar in approach to that used in [3]. Plane hypotheses are generated from minimal sets of points randomly sampled from the subset of point features with variance $\sigma^2_{max} < \sigma^2_T$, where σ^2_{max} is the maximum of the variances along each dimension and σ_T is a suitably chosen threshold. Each hypothesis is tested for consensus with the rest of the set. A minimal set of point features $(\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3)$ generates the parameters of the plane hypothesis as follows:

$$\mathbf{p}_o = \mathbf{m}_1 \quad \mathbf{c}(\theta_1, \phi_1) = \mathbf{m}_2 - \mathbf{m}_1 \quad \mathbf{c}(\theta_2, \phi_2) = \mathbf{m}_3 - \mathbf{m}_1 \tag{7}$$

and a point $\mathbf{m}_i$ is deemed to be in consensus with the hypothesis if its perpendicular distance from the plane, d, is less than d_T, where $d = (\mathbf{m}_i - \mathbf{p}_o) \cdot \mathbf{n}$, and its Euclidean distance from the plane origin is less than d_{max}. The second test ensures that we only initialise planes with strong local support. This is important because it is known that the relative positions of nearby points in a SLAM system are typically very accurate, even if the uncertainty in the global positions is large, so introducing a plane using a set of local points is relatively safe. However, we can make no such assumptions about the relative positions of widely dispersed points in the SLAM map.

The best-fit plane is determined from the inlying point features for the plane hypothesis with most consensus. The origin is set to the mean and the orientation parameters are determined from the principal components. Specifically, if the position vectors for l inlying points w.r.t the mean are stacked into an $l \times 3$ matrix $\mathbf{M}$, then the eigenvector of $\mathbf{M}^T\mathbf{M}$ corresponding to the smallest eigenvalue gives the normal to the plane and the other two eigenvectors give the basis vectors within the plane. The smallest eigenvalue, λ_{min}, is the variance of the inliers in the normal direction and provides a convenient measure of the quality of the fit.

In order to avoid adding poor estimates of planes to the system state, the best-fit plane generated by the RANSAC process is only initialised in the SLAM system if $l > l_T$ and $\lambda_{min} < \lambda_T$. The best-fit parameters are used to initialise a plane feature in the state and the covariance is updated according to (3), where $\mathbf{s}(\mathbf{x}, \mathbf{z}_o)$ denotes the derivation of the plane parameters from the set of inlying point features. Thus, both $\mathbf{z}_o = \mathbf{0}$ and $\nabla \mathbf{s}_\mathbf{v} = \mathbf{0}$, whilst the Jacobian w.r.t an inlying point feature $\mathbf{m}_i$ is computed as:

$$\nabla \mathbf{s}_{\mathbf{m}_i} = \left[\frac{\partial \mathbf{p}_o}{\partial \mathbf{m}_i} \ , \ \frac{\partial (\theta_1, \phi_1)}{\partial \mathbf{c}(\theta_1, \phi_1)} \frac{\partial \mathbf{c}(\theta_1, \phi_1)}{\partial \mathbf{m}_i} \ , \ \frac{\partial (\theta_2, \phi_2)}{\partial \mathbf{c}(\theta_2, \phi_2)} \frac{\partial \mathbf{c}(\theta_2, \phi_2)}{\partial \mathbf{m}_i} \right]^T \tag{8}$$

where $\partial \mathbf{c}(\theta_1, \phi_1)/\partial \mathbf{m}_i$ and $\partial \mathbf{c}(\theta_2, \phi_2)/\partial \mathbf{m}_j$ are the Jacobians of the two eigenvectors computed using the method described in [16]. This ensures that the plane parameters

are correlated with the rest of the SLAM state through the inlying point features. However, we do not immediately convert these inlying feature points to 2-D points on the plane in case the latter proves to be an unstable addition to the map. Instead, we choose to gradually add points to the plane using the mechanism described in the next section.

3.4 Adding and Fixing Points in Planes

At each time step, all converged 3-D point features in the map are considered as candidates for transformation to 2-D points associated with planes augmented to the map. A 3-D point feature is transformed using (6) if $\sigma^2_{max} < \sigma^2_T$ and its perpendicular distance from the plane and its distance from the plane origin are within d_T and d_{max}, respectively. The state covariance is then updated using (4). Since the new representation is dependent on the plane parameters, this process introduces correlations with the other 2-D points associated with the same plane.

If the maximum of the variances of a 2-D planar point becomes very small, less than a threshold σ_{fix}, then we can consider a further collapse of the state space by removing it completely. Instead of maintaining an estimate of its 2-D position in the plane, we consider it as a fixed point in the plane and use its measurements to update the associated planar feature directly (as described in the next section). This leads to a significant reduction in the size of the state space but does so at the cost of introducing errors and inconsistency into the system (c.f. Section 4).

3.5 Plane Measurement Equation

It is not possible to make a direct observation of the plane. However, each observed feature on the plane imposes a constraint between the camera pose, the position and orientation of the plane and the position of the observed 2-D point. The measurement model for a 2-D planar point $\mathbf{m}_i$ is very similar to that of a standard 3-D point feature, but involves an additional preliminary step to convert it to a 3-D position vector $\mathbf{p}$ in the world frame of reference prior to projection into the camera, i.e. from (6)

$$\mathbf{p} = [\mathbf{c}(\theta_1, \phi_1) \;\; \mathbf{c}(\theta_2, \phi_2)]\, \mathbf{m}_i + \mathbf{p}_o \tag{9}$$

The predicted measurement for the planar point can then be obtained by passing $\mathbf{p}$ through the standard perspective projection model in (2). The similarity between the measurement models makes the implementation of planar points in the existing EKF SLAM system very simple. The measurement Jacobian required by the EKF is much the same as that for 3-D point features, except that we need to take account of the conversion in (9) when computing the Jacobian relating the observation to 2-D point feature, i.e

$$\frac{\partial \mathbf{h}_i}{\partial \mathbf{m}_i} = \frac{\partial \mathbf{h}_i}{\partial \mathbf{p}} \, \frac{\partial \mathbf{p}}{\partial \mathbf{m}_i} \tag{10}$$

where $\mathbf{h}_i$ is the measurement function associated with a 3-D point (eqn 2) and $\partial \mathbf{p} / \partial \mathbf{m}_i$ is the Jacobian of the 3-D point w.r.t the 2-D planar point, derived from (9). Finally, since the predicted observation is also dependent on the position and orientation of the plane, a Jacobian $\partial \mathbf{h}_i / \partial \mathbf{m}_j$ relating $\mathbf{h}_i$ to the relevant plane feature $\mathbf{m}_j$ is also derived from (9).

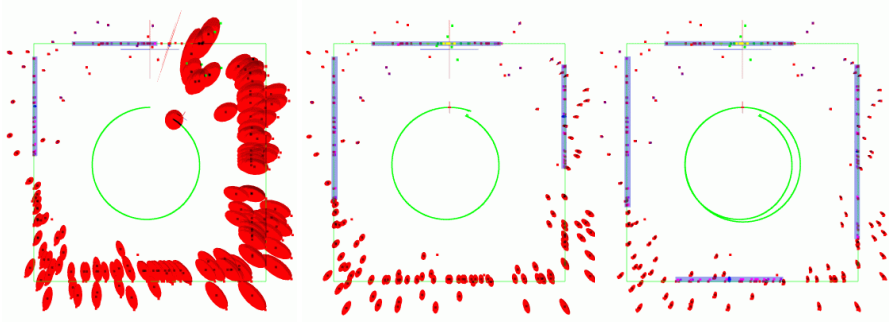


Figure 1: Simulation results: (left) large covariance just prior to loop closure; (middle) covariance and map error reduction after loop closure; (right) further reductions in covariance and discovery of additional plane on second loop.

4 Experiments

Experiments were carried out to test the effectiveness of plane discovery and the effect of collapsing the state space on the consistency of the SLAM estimation. Simulated data was used to determine the accuracy of tracking and reconstruction against a known ground truth. Performance was also assessed for real-time operation using an agile hand-held camera within an office environment.

The simulation models a 6 d.o.f. camera moving through an environment containing 200 3-D points arranged in a square room with walls 4m long (Fig. 1) and with 50% of the points lying at random positions on the walls. Newly observed points are initialised as 6-D inverse depth points and converted to 3-D points once their depth uncertainty becomes Gaussian [6]. Camera specifications are 43° horizontal field of view (FOV) and image size 320×240 pixels. Point features are observed with perfect data association and zero-mean Gaussian measurement error of 1.0 pixel^2 variance. The camera moves on a fixed circular trajectory of radius 1m around the room centre and maintains a radial orientation. Each simulation was run for two full loops of the camera trajectory. We used the following thresholds, chosen to minimise mismatches of points to planes: $d_T = 0.5cm$, $l_T = 7$, $\lambda_T = d_T^2$, $d_{max} = 200cm$. Varying these values produces different numbers and sizes of planes. The simulations were run for three different SLAM scenarios: with points only; with inference of planes; and with inference of planes and fixed planar points. Different values for the thresholds on point feature variance were tested: a strong threshold $\sigma_T = 2.0cm$, $\sigma_{fix} = 0.1cm$ (cases B and D in Fig. 2); a weak threshold $\sigma_T = 10.0cm$, $\sigma_{fix} = 1.0cm$ (cases C and E); and a medium threshold $\sigma_T = 10.0cm$, $\sigma_{fix} = 0.1cm$ (case F).

Figure 1 shows an external view of the camera position, trajectory, and map estimates prior to and following loop closure. The ellipses indicate position and point variances and the plane estimates are shown superimposed on the ground-truth outline of the room. In this case, a strong threshold was used for plane discovery and planar points were not fixed. Note the accuracy of the plane estimates and the clear reduction in point covariance and correction of the camera trajectory after loop closure. This demonstrates that inference of higher level structure and collapsing of the state space can be achieved without losing consistency. Reductions in state size for the different simulations are shown in

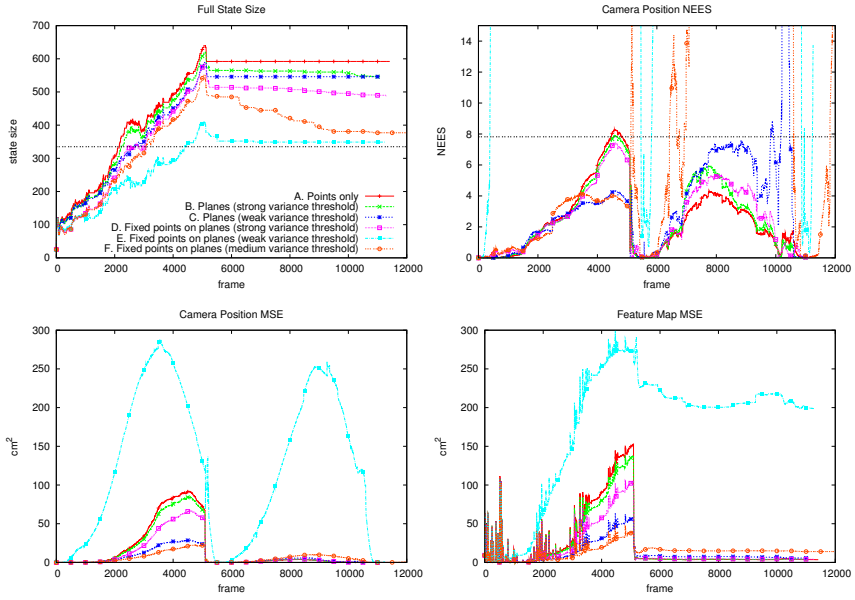


Figure 2: Simulation results: (top-left) total state size, horizontal line indicates minimum size if all planar points are added as fixed points; (top-right) NEES of camera position, horizontal line indicates consistency threshold; (bottom-left) MSE of camera position; (bottom-right) MSE of map features. Loop closure occurs in frames 5400 and 10800.

Fig. 2 (top-left). We observe that the different methods all lead to different levels of state reduction and the merits of each approach can be assessed by comparing the consistency and accuracy shown in the remaining plots of Fig. 2. Consider case E which achieves a large reduction in state size but becomes inconsistent and with large mean-squared error (MSE). In contrast, case D achieves savings in state size compared to points only (case A) but maintains consistency. The consistency of the camera position estimates (Fig. 2 (top-right)) was measured using the Normalised Estimation Error Squared (NEES) [2] function $(\mathbf{v}-\hat{\mathbf{v}})^{\mathrm{T}}\hat{\mathbf{P}}_{\mathbf{vv}}^{-1}(\mathbf{v}-\hat{\mathbf{v}})$. The figure also shows the threshold $\chi^2_{r,1-\alpha}$ from the χ^2 distribution with $r = dim(\mathbf{v}) = 7$ degrees of freedom and α the desired significance level (0.95 in our experiments). Note that consistency is strongly affected when we try to aggressively fix planar points to reduce state size (cases C, E and F). However, more conservative approaches (cases B and D) achieve useful state space reductions and introduce higher order structure without adversely affecting consistency.

MSE analysis of camera and feature position estimates (Fig. 2 (bottom)) shows the cost of introducing inconsistency. The most inconsistent methods do not benefit fully from loop closure and are unable to correct the error in their maps. More conservative thresholds maintain consistency and achieve similar accuracy to the points only approach with the additional benefit of reduced state size. There is also an indication that the MSE during the first loop is reduced when we add planes to the system (cases B, C, D and F). This may be because a good plane estimate extends its influence further into the map than a good point, due to its higher level structure, and any points constrained to lie on

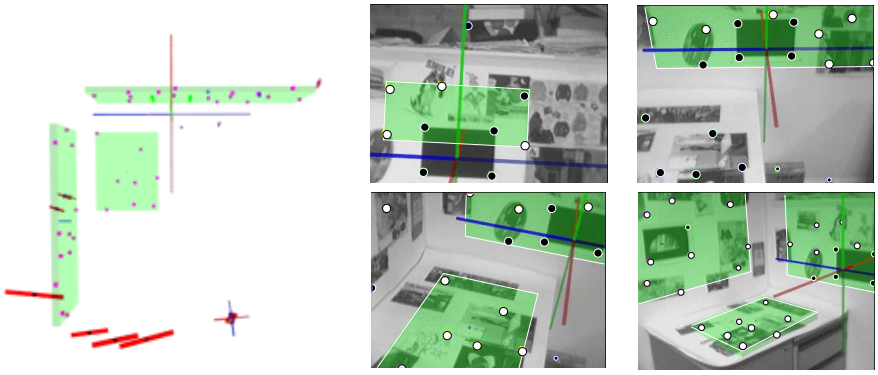


Figure 3: Real-time operation with hand-held camera: (left) view of camera position, map estimates and planar features; (middle and right) planar features (boundaries in white) are discovered with converged 3-D points in black and 2-D planar points in white. Videos available at this paper's entry at `www.cs.bris.ac.uk/Publications`.

this plane will also have their accuracy improved. This effect could be exploited by using known planes to initialise the SLAM system.

We also tested the method in real-time in an office environment using a calibrated hand-held web-cam with a resolution of 320×240 pixels and 43° FOV. Tracking was initialised with four known map points corresponding to the corners of a planar black on white rectangle placed in the scene. The environment contained three distinct planar surfaces with sufficient texture to encourage reliable feature detection and recognition. The former was based on a fast salient operator [17] and the latter on multiscale SIFT-like descriptors with scale prediction [5]. This gives robust performance, even in the face of significant erratic camera movement. Figure 3 shows an external view of the camera position and map estimates part way through the sequence and views through the camera with mapped point and planar features superimposed. Note that the three planes in the scene have been successfully detected and the vast majority of the 3-D points mapped were transformed into 2-D planar points (except those on the calibration pattern which for fair comparison we force to remain as 3-D points). In this experiment, the system operated at around 20 frames/sec, including graphics rendering and without software optimsation, for maps of up to 50 features and with 10 visible features per frame on average.

5 Conclusions

Augmenting the SLAM state with parameters representing a planar structure allows us to represent mapped points lying on the plane with a reduced parameterisation and collapse the state space whilst maintaining a consistent, full-covariance representation. This technique can be applied in real-time visual SLAM systems to reduce computational demands, as well as giving a higher level scene description suitable for further applications. Future work will consider the incorporation of a wider variety of structural information, such as lines and junctions. Knowledge of planar structure in the scene may also have benefits for visual feature detection and matching which can be exploited to improve performance.

Acknowledgements

This work was funded by the EPSRC Equator IRC and ORSAS UK.

References

[1] T. Bailey and H. Durrant-Whyte. Simultaneous localisation and mapping (slam): Part ii - state of the art. *IEEE Robotics and Automation Magazine*, (3), 2006.

[2] Y. Bar-Shalom, T. Kirubarajan, and X.R Li. *Estimation with Applications to Tracking and Navigation*. 2002.

[3] A. Bartoli. A random sampling strategy for piecewise planar scene segmentation. *Computer Vision and Image Understanding*, 105(1):42–59, 2007.

[4] R. O. Castle, D. J. Gawley, G. Klein, and D. W. Murray. Towards simultaneous recognition, localization and mapping for hand-held and wearable cameras. In *Proc. Int. Conf. Robotics and Automation*, 2007.

[5] D. Chekhlov, M. Pupilli, W. Mayol, and A. Calway. Robust real-time visual slam using scale prediction and exemplar based feature description. In *Proc. Int Conf on Computer Vision and Pattern Recognition*, 2007.

[6] J. Civera, A.J. Davison, and J.M.M. Montiel. Inverse depth to depth conversion for monocular slam. In *Proc. Int. Conf. Robotics and Automation*, 2007.

[7] Andrew J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proc. Int. Conf. on Computer Vision*, 2003.

[8] H. Durrant-Whyte and T. Bailey. Simultaneous localisation and mapping (slam): Part i the essential algorithms. *IEEE Robotics and Automation Magazine*, (2), 2006.

[9] E. Eade and T. Drummond. Edge landmarks in monocular slam. In *Proc. British Machine Vision Conf*, 2006.

[10] E. Eade and T. Drummond. Scalable monocular slam. In *Proc. Int Conf on Computer Vision and Pattern Recognition*, 2006.

[11] M.A. Fischler and R.C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[12] A.P. Gee and W. Mayol-Cuevas. Real-time model-based slam using line segments. In *Int. Symp. on Visual Computing*, 2006.

[13] W.W. Mayol, A.J. Davison, B.J. Tordoff, N.D. Molton, and D.W. Murray. Interaction between hand and wearable camera in 2d and 3d environments. In *Proc. British Machine Vision Conf*, 2004.

[14] N. Molton, I. Ried, and A.J Davison. Locally planar patch features for real-time structure from motion. In *Proc. British Machine Vision Conf*, 2004.

[15] J.M.M. Montiel, J. Civera, and A.J. Davison. Unified inverse depth parametrization for monocular slam. In *Robotics: Science and Systems Conf.*, 2006.

[16] T. Papadopoulo and M.I.A. Lourakis. Estimating the jacobian of the singular value decomposition: Theory and applications. In *Proc. Euro Conf on Computer Vision*, 2000.

[17] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *Proc. Euro Conf on Computer Vision*, 2006.

[18] P. Smith, I. Reid, and A.J. Davison. Real-time monocular slam with straight lines. In *Proc. British Machine Vision Conf.*, 2006.

[19] R. Smith, M. Self, and P. Cheeseman. A stochastic map for uncertain spatial relationships. In *Proc. Int. Symp. Robotics Research*, 1987.

[20] B. Williams, P. Smith, and I. Reid. Automatic relocalisation for a single-camera simultaneous localisation and mapping system. In *Proc. Int. Conf. on Robotics and Automation*, 2007.

A Probabilistic Framework for Recognizing Similar Actions using Spatio-Temporal Features

Alonso Patron-Perez, Ian Reid
Department of Engineering Science, University of Oxford
OX1 3PJ, Oxford, UK
{alonso,ian}@robots.ox.ac.uk

Abstract

One of the challenges found in recent methods for action recognition has been to classify ambiguous actions successfully . In the case of methods that use spatio-temporal features this phenomenon is observed when two actions generate similar feature types. Ideally, a probabilistic classification method would be based on a model of the *full joint* distribution of features, but this is computationally intractable. In this paper we propose using an approximation of the full joint via first order dependencies between feature types using so-called Chow-Liu trees. We obtain promising results and achieve an improvement in the classification accuracy over naive Bayes and other simple classifiers. Our implementation of the method makes use of a binary descriptor for a video analogous to one previously used in location recognition for mobile robots. Because of the simplicity of the algorithm, once the offline learning phase is over, real-time action recognition is possible and we present an adaptation of this method that works in real-time.

1 Introduction

The interpretation of video sequences is a topic that has been growing in importance in recent years. It involves not only computer vision techniques for obtaining information from images but also machine learning algorithms because of the ultimate need for high level interpretation. Many things are involved in the analysis of a video sequence such as object recognition, target localization, motion detection, physical contraints and human behaviour. In some applications, like visual surveillance, the role played by humans is the main priority, so several methods for modeling human activity have been proposed. However the majority of these approches fall short of modelling "behaviour", which, we posit, refers to a more complex combination of factors like intentionality and therefore belongs to a higher level of understanding.

A first step to achieving behaviour undestanding is almost certainly the recognition of actions. In Efros *et al.* [8] a target motion descriptor was developed for classifying frames using a *k*-nearest neighbour classifier. Robertson [18] used Efros' descriptor within a probabilistic framework, fusing the low-level motion descriptors with location and direction cues via a Bayes' net. That work further made an attempt at behaviour recognition via

hidden Markov representations of action sequences which they proposed could be interpreted as behaviours.

Another kind of descriptor for action classification was employed by Davis [6], obtained as histograms of the motion orientation information contained in what he called "Motion History Images" (MHI). Carlsson and Sullivan [3] utilized keyframes to represent actions: given a new frame a voting matrix is created to compare keyframes with the new frame, with the matrix updated if the topology between corresponding points of a keyframe and the new frame is the same. An interesting approach was followed by Boiman and Irani [2], although their main objective was to detect irregularities in videos and not to classify actions. They define irregularities as fragments of videos that can't be composed using previous observed data. The method apparently works well but is computationally expensive.

Other methods developed lately are based directly on image features. Image features are a common tool in computer vision methods, very popular in areas such as pattern and object recognition. In recent years a new kind of feature, found not only in space but also in time, have been used for action recognition. Laptev and Lindeberg [12] presented an approach that is an extension in time of the well know Harris-Stephens [9] corner detector, achieving good results in general although for subtle movements (like a wheel spinning) the method doesn't give a good response and the number of features obtained is very small. Oikonomopoulos *et al.* [17] also extended another kind of spatial features to the temporal case, in this case the saliency approach of Kadir and Brady [10]. The saliency approach is based on localizing points of local maximum entropy. A method that is not an extension of a previously know feature detector was presented by Dollár *et al.* [7], based primarily on a convolution with a Gabor filter in time. The features obtained in this way have also been used for action recognition by Niebles *et al.* [16] in a very interesting application of probabilistic Latent Semantic Analysis.

In this paper we adopt the spatio-temporal features proposed by Dollár *et al.*. In constrast to their work, however, our classification method is solidly grounded in a probabilistic framework. More precisely we are interested in evaluating $p(z_1, \ldots, z_n|A)$, the joint distribution over feature observations z_i given a particular action A. This joint distribution, comprising n dimensions (where n is typically very large) is intractable to compute. Our main contribution in this paper is to show how this joint distribtion can be effectively approximated using only first-order dependencies [4]. This idea was recently employed in mobile robotics for location recognition [5] and here we apply it to action recognition for the first time.

The remainder of the paper is divided as follows: in section 2 we review the theory behind spatio-temporal features and explain why similar actions are more dificult to identify following this approach. In section 3 our probabilistic method is explained in detail, and in section 4 we describe a real-time implementation of this method. Section 5 shows the results obtained and the last section deals with the conclusions and future directions of research.

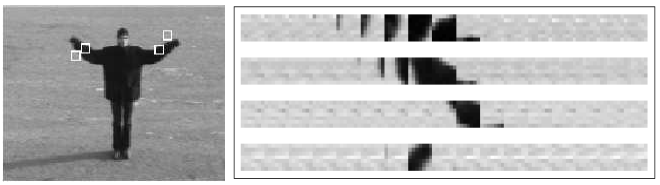
2 Spatio-temporal features

We will use the spatio-temporal features as described in [7], because they have shown good results even using the most simple methods of classification. These features are

obtained by convolving a set of video frames with a Gaussian $g(x,y;\sigma)$ in the spatial dimension and with 1D Gabor filters in the temporal dimension. The 1D Gabor filters are defined as $h_{ev}(t;\tau,\omega) = -\cos(2\pi t\omega)\exp^{-t^2/\tau^2}$ and $h_{od}(t;\tau,\omega) = -\sin(2\pi t\omega)\exp^{-t^2/\tau^2}$. In this quadrature pair each filter is a harmonic function with a Gaussian envelope. The parameter τ in this specific case controls the frequency of the harmonic function and the variance of the Gaussian. The whole response function is then expressed as:

$$R = (I * g * h_{ev})^2 + (I * g * h_{od})^2 \tag{1}$$

Using a threshold and non-maximal suppression, the features with highest response are selected. Usually these high response locations correspond to local regions containing periodic motions but regions with spatio-temporal corners [12] are also detected. Once the spatial and temporal location of a feature is obtained, the set of pixels which surround it in a given size of spatio-temporal window is extracted. [7] call this spatio-temporal window a *cuboid*. The side length of this temporal window is set to six times the scale at which the feature was detected. An example of the features detected by this method and their corresponding cuboids is shown in Figure 1.

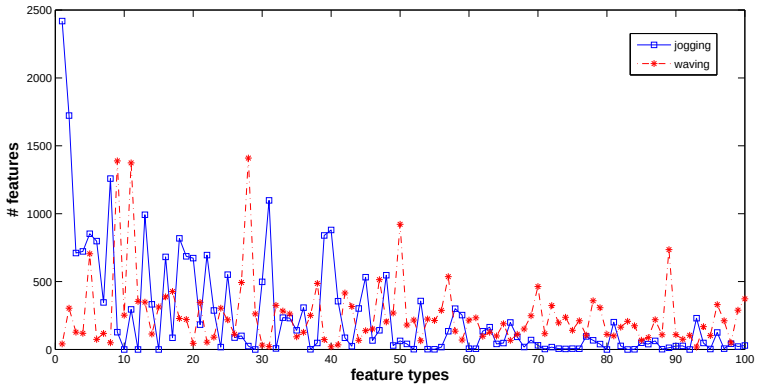


(a) (b)

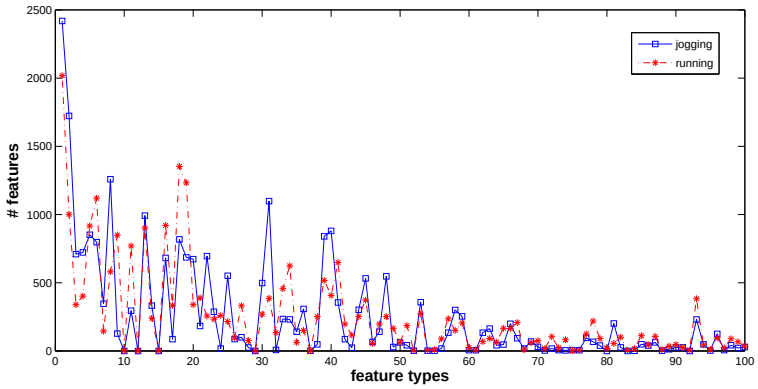
Figure 1: (a) Detected features. (b) Cuboids (flattened representation, time runs from left to right).

As shown in [14] the selection of a good descriptor for spatial features has proven to play an important role in improving the results of any algorithm involving image features. Regarding spatio-temporal features, [7] experiments with some of the classic descriptors including SIFT [13], and show that apparently a simple descriptor such as the concatenated brightness values of the cuboids gives results as good as or better than other descriptors. We have not conducted our own evaluation and like [7] we use this simple descriptor in our work. The features are clustered using *k*-means with Euclidean distance and a fixed number of cluster centres. The clusters centres obtained in this process define the set of feature types. After the clustering process each feature is tagged as belonging to one of these clusters. The overall process involves the selection of four parameters: the scales in space and time (σ,τ), the feature threshold, and the number of cluster centres. The latter determines the complexity of our model.

Using feature types for classifying similar actions has its problems because similar actions generate similar features as can be observed in Figure 2. This is what motivates the development of our method that can deal to some extent with this problem.



(a)



(b)

Figure 2: Graphics of feature ocurrence for (a) Jogging and Waving. (b) Jogging and Running. As can be seen similar actions, like those in (b), produced similar kinds of features.

3 Probabilistic formulation

Different methods have been used for the classification of actions. In [7] histograms of feature types using the Euclidean and χ^2 distances are compared. Laptev and Lindeberg [12] define models for each action and select the model that best fits the features. Probabilistic methods are used in [17] and [16]. The former uses Relevance Vector Machines for obtaining posterior probabilities of actions, and in the latter a probabilistic Latent Semantic Analysis is implemented.

Our implementation follows the spirit of these latter two approaches, recognizing that probabilistic approaches are more robust and give more relevant information that can be

used in higher levels of understanding. The approach described here is based on the method proposed by Cummins and Newman [5]. In their work the aim was to recognise a previously visited location for the specific case of loop closing in robot navigation. It is a relatively simple probalistic approach, but as we will show gives remarkably good results.

Our aim is to classify a *video* as one of the set of previously known actions. Here a video can refer to any sequence of frames, but in the case of our training data (the KTH dataset [11]) each video is in fact a single file, and we have one action per file. In our real-time implementation in section 4 we employ a sliding temporal window with the "video" being the sequence of frames within the window. We begin defining an observation vector as $Z = \{z_1, z_2, ..., z_n\}$ where each z_i is a binary variable that is set to 1 if a feature type is observed in a video and zero otherwise. Defining a set $A = \{A_1, ..., A_k\}$ of actions we want to obtain the conditional probability $p(A_i|Z)$ for each action. Using Bayes rule this is:

$$p(A_i|Z) = \frac{p(Z|A_i)p(A_i)}{p(Z)} \tag{2}$$

where $p(Z)$ is just a normalizing factor. Given the prior that each action is equally likely to occur, the important term in (2) becomes the likelihood: $p(Z|A_i)$. This conditional distribution contains the information on the relationships of the feature types and is therefore the one that we want to learn. Because the complexity of the calculation of the joint distribution increases exponentialy with the number of variables (feature types), computing of the full distribution quickly becomes untractable. We believe that capturing at some extent these relationships between feature types is a key factor for differentiating similar actions, therefore a way of approximating this distribution is of paramount importance. In the next sections we describe an approximation given by Chow and Liu [4] and also we briefly describe the naive Bayes approach as a standard method for comparison.

3.1 Naïve Bayes

One of the simplest ways of approximating a joint distribution is to assume independence between all the variables; this is called naïve Bayes and is expressed mathematically as $p(x_1, x_2, ..., x_n) = \prod_i^n p(x_i)$, in our case this will be:

$$p(Z|A_j) = p(z_1, z_2, ..., z_n|A_j) = \prod_i^n p(z_i|A_j) \tag{3}$$

This is a very simple way of approximating the joint distribution and very easy to implement. We simply have to learn the marginal probability of each feature type, and this can be done as explained in [15] by setting $p(z_i|Aj) = n(z_i)/N_j$, where $n(z_i)$ is the number of times that feature type z_i was observed and N_j is the total number of videos corresponding to action j. This kind of estimation has one drawback: if a feature type is not observed during training then its marginal probability becomes zero and therefore the whole joint becomes zero. To avoid this extreme case we apply a simple way of smoothing namely $p(z_i|Aj) = (n(z_i) + s)/(N + s * v)$ where s is the smoothing factor and v is the number of values that the random variable z can take. If $s = 1$ this is known as Laplacian smoothing.

3.2 Chow - Liu approximation

Chow and Liu [4] presented a first order approximation to the full joint distribution (i.e. the joint is represented as the product of terms $p(z_i|z_j,A)$, so that each observation variable z_i is conditioned on at most one other z_j). The distribution obtained in this way is proved to be the optimal approximation of the full distribution in the sense of minimizing the Kullback-Leibler divergence.

In order to determine the approximation the mutual information I between pairs of variables is calculated

$$I(z_i,z_j) = \sum_{z_i,z_j} p(z_i,z_j) \log \frac{p(z_i,z_j)}{p(z_i)p(z_j)} \tag{4}$$

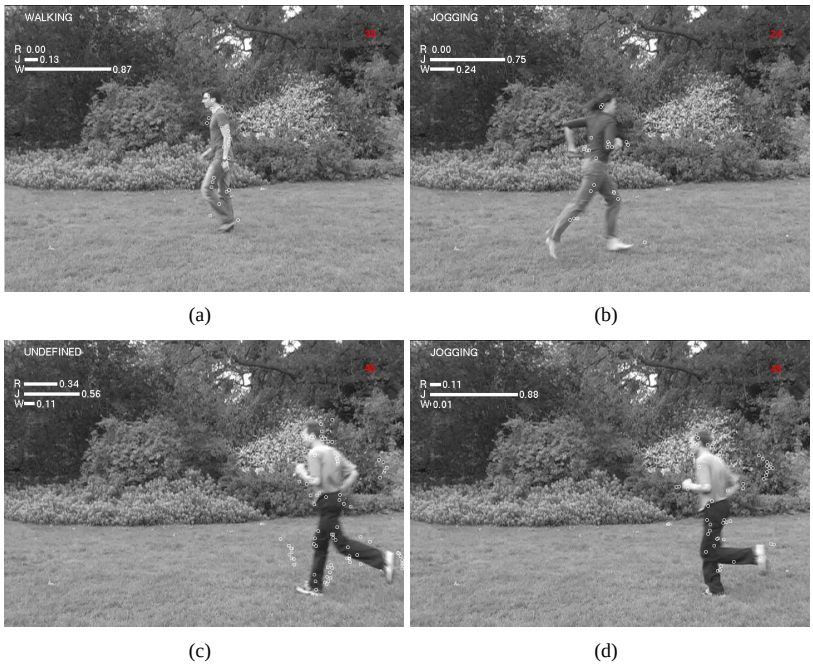
where the sum is made for all combination of values that the variables z_i and z_j can take. The joint distribution $p(z_i,z_j)$ and the marginals $p(z_i)$, $p(z_j)$ can be learnt from data. The next step is to construct a weighted undirected graph whose nodes are the random varibles z_i and the weight of each edge is given by the mutual information between its two nodes. The final step involves finding the maximum spanning tree in this graph, as task common in graph theory. The maximal spaning tree provides the structure of the dependencies between the variables: each variable is dependent on only one other variable (except for the root of the tree). The approximation of the full distribution is then

$$p(Z|A_j) = p(z_r|A_j) \prod_{q\in\Omega} p(z_q|z_{p_q},A_j) \tag{5}$$

where z_r is the root node, Ω is the set of all nodes excluding the root node and z_{p_q} is the parent of node z_q. As in the case of naïve Bayes we have to apply some kind of smoothing to avoid probabilities going to zero.

4 Real-Time implementation

Having obtained action descriptors in an offline learning phase, the recognition task is extremely fast. For our real-time implementation we maintain a FIFO buffer of video frames in memory. The size of this buffer of course depends on the temporal scale chosen and this induces a fixed latency of half the buffer length on the results. Each image received is first smoothed with a Gaussian, and then inserted into the buffer. Once the buffer is full we need only perform the convolution in time, extract the features of the central frame in the buffer, and compute a binary frame descriptor for that frame in an analogous fashion to the complete video in section 3. We then estimate the posterior probability for each one of our actions given this new data. If the estimated probability for the maximum likelihood action is above a threshold then we identify the current action, and if not we read another frame and add its new frame descriptor to our previous one and compute the probabilities again, this continues until the probability of one of the actions reaches the threshold. Frames older than a fixed lag are removed from consideration. Images of the real-time system working are shown in Figure 3.



(a) (b)

(c) (d)

Figure 3: Snap-shots of the real-time implementation. (a)-(b) Correct classification of the action. (c) In this frame there is confusion between jogging and running so the action remains undefined until more information is acquired. Two frames later, in image (d), the action is correctly classified.

5 Experiments

For our experimental evaluation we have used the action database made by Laptev and that is available online at KTH website [11]. This database contains videos of six different actions namely: hand clapping, hand waving, walking, jogging, running and boxing, performed by 25 different persons and has been used in various papers as their training and test data. The total database contains 599 videos. Because our main objective is to compare the results obtained by the Chow-Liu approximation when trying to classify similar actions against the simple naive Bayes approach we first tested with three actions: walking, jogging and running. A set of images extracted from the videos corresponding to these three actions can be seen in Figure 4.

The methodology chosen for the experiments is the one used in [16]. The database was divided in 25 sets, one for each person, we used the features extracted from the videos of 3 randomly selected persons for clustering (and obtain our feature types). With the remaining 22 persons we performed a leave-one-out cross validation test. Because the clustering phase introduces a random component, the experiments were repeated 25 times for each number of cluster centres and the results presented here are an average of these runs. The parameters used for the feature detection were $\sigma = 1$ (spatial scale), $\tau = 2.5$

Figure 4: Some images taken from the videos of the KTH database.

and a threshold of 0.01. The scales were fixed during the experiments and we only used the first 200 frames of each video for extracting features. The results of both the naïve Bayes approach and the Chow-Liu approximation for different cluster centres, can be seen in Figure 5a. Figure 5b shows the confusion matrix obtained with the Chow-Liu algorithm and 800 cluster centres. It should be noted that in measuring the performance of discrimination between actions (especially jogging and running) we take the labels in the KTH dataset as ground truth, but these labels are very much open to individual interpretation. In order to asses this ambiguity in the "ground-truth" we asked several people to classify 300 videos of walking, jogging and running. The confusion matrix of the average results obtained in this test is shown in Figure 5c.

In order to compare our classification scheme with the results presented in other papers we also made experiments using the whole database. Figure 6 show the results of both methods when training and testing with the six actions using 500 cluster centres.

6 Conclusions

We have presented an action classification method that exploits the relationships between different spatio-temporal feature types and we have shown it to improve the classification of similar actions. The results are promising despite some ambiguity of the data set ground truth. We also showed how this method can be implemented in a real-time application. It is also worth remarking that even the simple naïve Bayes approach gives very good results when classifying disimilar actions, as can be seen in the results shown in Figure 6, indeed the accuracy for these actions is sometimes better that the one obtained using more complex models.

Although in this paper we have used a supervised learning approach, this can be easily extended to an usupervised form in the case of naïve Bayes by means of Mixtures of Bernoulli distributions and EM as described in [1]. The case of an unsupervised Chow-Liu algorithm is more difficult and the subject of our ongoing research. In common with the related work of [7], currently our overall action descriptor – the binary occurrence vector – takes no account spatial or temporal ordering, factors which we believe may be

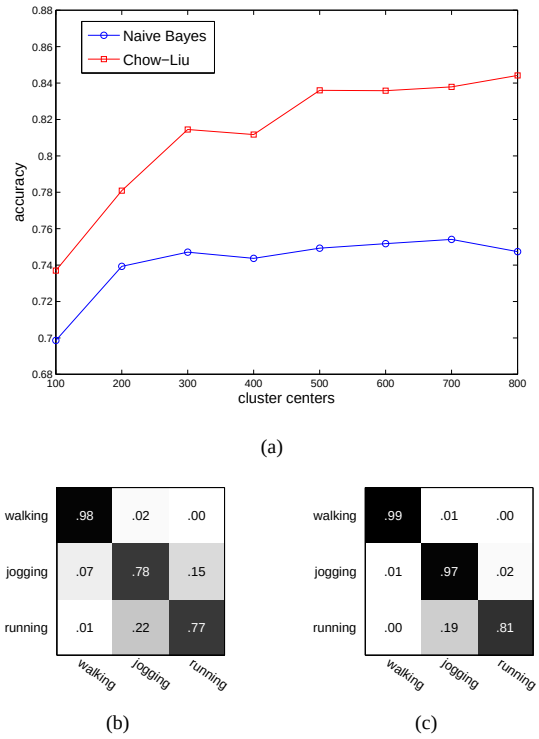


Figure 5: Results obtained. (a) Accuracy of both methods using different cluster centres. (b) Confusion matrix obtained with the Chow-Liu algorithm and 800 cluster centres. (c) Confusion matrix as a result of human classification.

important to incorporate into our descriptor.

7 Acknowledgements

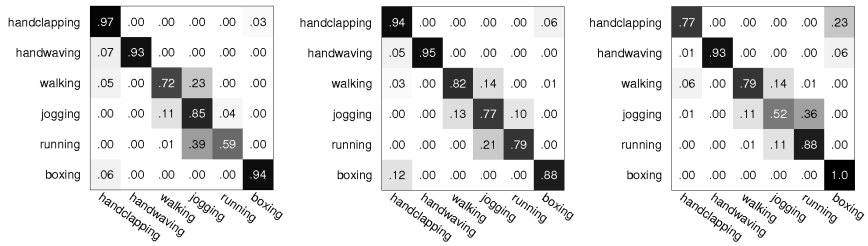
We would like to thank Piotr Dollár for allowing us the use of his code. This work was supported by CONACYT and the EU FP6 project *HERMES*.

References

[1] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[2] O. Boiman and M. Irani. Detecting irregularities in images and in video. In *International Conference on Computer Vision*, volume 1, pages 462–469, 2005.

[3] S. Carlsson and J. Sullivan. Action Recognition by Shape Matching to Key Frames. In *Workshop on Models versus Exemplars in Computer Vision*, 2001.



(a) (b) (c)

Figure 6: Confusion matrices for the six actions obtained using 500 cluster centres: (a) naïve Bayes. (b) Chow-Liu. (c) Results obtained by Niebles et al. [16] (unsupervised learning).

[4] C.K. Chow and C.N. Liu. Approximating Discrete Probability Distributions with Dependence Trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968.

[5] M. Cummins and P. Newman. Probabilistic Appearance Based Navigation and Loop Closing. In *International Conference on Robotics and Automation*, 2007.

[6] J. Davis. Recognizing Movement using Motion Histograms. Technical Report 487, MIT Media Lab Perceptual Computing Group, MIT, 1999.

[7] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, pages 65–72, 2005.

[8] A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing Action at a Distance. In *International Conference on Computer Vision*, Nice, France, 2003.

[9] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–151, 1998.

[10] T. Kadir and M. Brady. Scale Saliency: A Novel Approach to Salient Feature and Scale Selection. In *International Conference on Visual Information Engineering*, pages 25–28, 2003.

[11] I. Laptev and B. Caputo. Recognition of human actions. Action Database. http://www.nada.kth.se/cvap/actions/, 2007.

[12] I. Laptev and T. Lindeberg. Space-time Interest Points. In *International Conference on Computer Vision*, 2003.

[13] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 2004.

[14] K. Mikolajczyk and C. Schimd. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[15] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[16] J. Niebles, H. Wang, and L. Fei-Fei. Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words. In *British Machine Vision Conference*, 2006.

[17] A. Oikonomopoulos, P. Ioannis, and M. Pantic. Spatio-Temporal Salient Points for Visual Recognition of Human Actions. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 36(3):710–719, 2006.

[18] N. Robertson and I. Reid. Behaviour understanding in video: a combined method. In *International Conference on Computer Vision*, 2005.

Associating People Dropping off and Picking up Objects

Dima Damen David Hogg
School of Computing, University of Leeds
dima@comp.leeds.ac.uk, dch@comp.leeds.ac.uk

Abstract

Several interesting monitoring applications concern people entering a prescribed area, where they deposit an object in their possession, or collect an object deposited earlier. One example arises in the use of bicycle racks. We propose a novel method for associating each person who deposits an object with the person who later collects it. Our main contribution is to deal with ambiguity in the visual data through the use of global constraints on what is possible. The method is evaluated on a set of practical experiments in a bicycle rack, and applied to online theft detection by comparing the colour profile of associated individuals.

1 Introduction

A major challenge in computer vision is to reliably recognise events based on current object tracking and detection technology. Despite intensive research aiming towards universal tracking with "one object per track and one track per object" [17], such a tracker is not yet available if a single viewpoint is used. Ambiguous visual analysis makes it difficult to associate each person with the event accomplished. Understanding the expected events and their underlying global constraints is one way to resolve conflicting and ambiguous observations.

The scenario where a person leaves an object (typically locked) within a storage area, and picks it up sometime later, presents a rich constrained scenario that may be observed by a single CCTV camera. However, ambiguous observations from available tracking and object detection methods are often insufficient to recognize the events in isolation with any certainty. We propose a method to connect people and objects, and decide on the sequence of drop-off and pick-up events. The explanations generated should be both consistent and optimal based on the observations. A link can then be inferred between the person dropping off an object, and the person picking up the same object later. Additionally, passive biometric features can be utilized to compare these two individuals, and raise a warning when they do not match.

In Section 2, we give a brief overview of relevant work in associating trajectories and biometric comparison. Section 3 describes how we detect people and objects from a video stream. Section 4 presents the association task and proposes a solution method. Three experiments using bicycles as objects were conducted and results are presented in Section 5. We show how solutions are superior to those obtained when the constraints are relaxed or removed altogether.

2 Background

Our scenario requires associating the individual who drops off an object with the one who picks it up. Establishing correspondence between temporally disjoint motion trajectories of individuals has previously been used to relate entry and exit points in non-overlapping camera views, and to track individuals across blind regions with a single view [2, 3, 9, 12, 22]. Such methods typically depend on features of the moving individuals obtained from the video stream, sometimes referred to as passive [6] or soft-biometrics [19, 20].

Colour is a matching feature to connect two trajectories as it is easy to retrieve, although depends on people not changing their clothing within a single session [3, 6, 7, 9, 16, 21]. Bowden and KaewTraKulPong utilized colour histograms to re-identify individuals in non-overlapping neighbouring camera views [3]. ZhiHua and Komiya also used colour and shape for pedestrians and vehicles [22]. Similarly, Berclaz et. al. used both colour and location information to fuse trajectories in overlapping cameras [2]. Sivic et. al., used colour similarity of clothing to match people in family photos segmented using a face detector [16].

There are however difficulties in using colour information of walking pedestrians. Shadows, lighting changes and clothing's natural folding introduce wide variations to the colour of the corresponding pixels across frames. A pixel-by-pixel comparison can thus produce poor matches. Wu et. al. showed how frame-level and sequence-level colour representations can overcome pixel-level variations [20]. Frame-level colour information is usually represented by Gaussian mixtures [8] or colour histograms [3, 7, 14, 16, 21]. Bowden and KaewTraKulPong proposed the median histogram of the per-frame histograms to represent sequence-level information [3].

In our scenario, we can not ensure the picking person to be the same as the dropping person. Thus, correspondence based on the individual's features can not be applied. Makris et. al. followed an approach that is independent of feature-based matching. Their work learns the temporal characteristics of the source-sink connection by observing regularities of exits and entries in different camera views, and thereby anticipating the topology of a set of camera views [12]. Javed et. al. combine temporal relationships with colour profiles and neighbourhood knowledge. They concentrate on learning brightness transfer functions to strengthen their colour matching scheme [9]. The temporal expectancy can not be used in our case either. We can not anticipate with any certainty when an object will be picked up.

Kettnaker and Zabih, on the other hand, tracked pedestrians between cameras using a one-to-one optimal assignment approach [10]. This is based on the valid assumption that a trajectory can be associated with only one trajectory that left a neighbouring camera. We experiment with a similar technique to link dropping off and picking up trajectories.

To the authors' knowledge, the exact proposed scenario of dropping off and picking up objects has not been explored before. Perhaps the closest scenario that has been addressed previously is that of abandoned baggage. This was the basis of the PETS2006 challenge for which a number of solutions were proposed [5]. The abandoned baggage task, however, does not require associating events as required in dropping off and picking up objects.

The method described in this paper uses sequential propagation of multiple hypotheses in order to solve a constrained optimisation problem. A similar technique has been used in the radar surveillance literature to deal with ambiguities introduced by sensors.

Reid proposed the Multiple Hypothesis Tracking (MHT) in 1979 for radar activities [15], where tracking information, availed sequentially, clears ambiguities in previous observations. MHT keeps a set of hypotheses explaining legal assignments in a tree structure. Each child hypothesis represents one interpretation of the new data. A path in the tree represents a possible interpretation of all observations from the beginning of the observation period (root) up to the current timestamp (leaf).

3 Tracking People and Detecting Objects

For tracking people, a generic blob tracker [11] was used to retrieve the trajectories of individuals as they approach and depart the storage area. This tracker uses a per-pixel background model together with a simple foreground shape model, and assigns a unique identifier to each object moving over a continuous trajectory. For each person, the position (represented by centre of mass of the pixels), area (number of foreground pixels) and colour information are provided for each frame during the period the person remains visible. We extended the tracker to deal with broken trajectories through combining trajectories that exhibit similar colour profiles and that are spatially and temporally consistent.

In the case where the possessed object is comparable to the individual's projected area (such as bicycles utilized in our experiments), the change in area along the trajectory is significant, and can be used to differentiate people depositing from those collecting objects. This is feasible when the viewed locations are at similar depth from the camera's position. Figure 1 shows projected area through time as people deposit and collect objects. The difference in the area prior to entering the storage area and after leaving it represents an estimate of the projected size of an object at a given depth, and is independent of the person's size. Maximum likelihood estimation (MLE) was used to estimate Gaussian class conditional densities for area differences (at a given depth) across a drop-off and a pick-up (Figure 1). In this way we are able to estimate the likelihood that someone is dropping off or picking up an object, provided a continuous trajectory is available across the event. We use this likelihood, when available, to help constrain the assignment of individuals to different events detailed in Section 4.

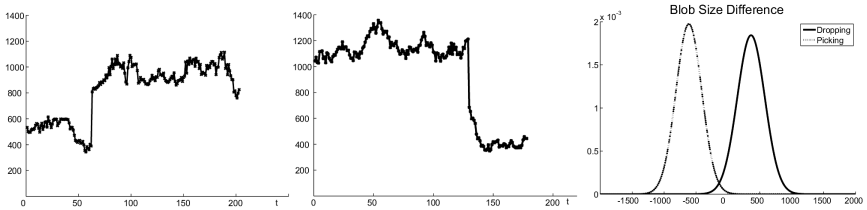


Figure 1: The change in area over time for someone picking (left) and dropping (middle) a bike; the increase/decrease in size is clearly visible around t=60 (left) and t=130 (middle), along with ML estimates for Gaussian distributions of area differences (right)

The tracker can not be used to identify static objects. Instead, 'before' and 'after' reference images of the storage area are compared, thereby revealing changed pixels, representing objects that have been deposited and removed. The risk of noise or lighting changes is minimised by taking reference images before a person enters the storage area

and after exiting it. If another person enters the area prior to the departure of the first, the second reference image is only taken after all have departed. We refer to these intervals as 'periods of activity'. The changed image pixels are then grouped into connected regions representing several blobs of dropped and picked objects as Figure 2 shows. We refer to these blobs as 'objects' even though some are actually multiple adjacent objects that could not be separated. The possibility of multiple objects of the same type within one detected blob is reflected in the constrained optimisation problem formulated in Section 4. We distinguish dropped from picked objects by detecting intensity edges in the 'before' and 'after' reference images, masking with the changed pixels, and classifying as a dropped object if the number of edge features increases, and vice versa. This assumes the background is relatively free of edge features, and that only one type of object (either dropped or picked) is present within each blob.

(a) (b) (c) (d)

Figure 2: Before (a) and after (b) reference images, revealing dropped (c) and picked (d) bicycles

4 The Association Phase

4.1 The problem as constrained optimisation

From tracking and object detection, a set of person hypotheses $\{p_i\}$ and a set of object hypotheses $\{o_i\}$ are generated. Individuals may appear more than once in the set of people, and, crucially, the same object is normally detected twice, once when it is dropped, and once when it is picked. False positives are expected in both sets. The association problem can be represented using a graph as in Figure 3. It connects people to objects and dropped objects to picked objects. There are two types of edges: person-object edges and object-object edges. The aim is to identify those edges that best explain the observations. An example of a solution is shown as darker lines to the right of Figure 3. Four types of events are used to explain the observations over an extended period:

1. $pkdp(p_i, o_j, p_k, o_l)$: person p_i picks up object o_j, person p_k drops off object o_l, o_j and o_l are the same object,
2. $dp(p_i, o_j)$: person p_i drops off object o_j, which remains unpicked during the period,
3. $pk(p_i, o_j)$: person p_i picks up object o_j, which was not dropped off during the period,
4. $none(p_i)$: person p_i neither picks up nor drops off an object.

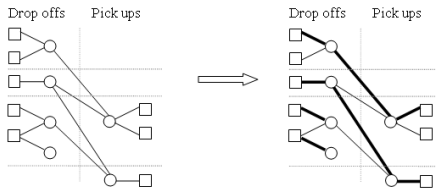


Figure 3: People (squares) and objects (circles) are connected via edges. Horizontal dividers separate periods of activity within the storage area, punctuated by periods of inactivity. The vertical divider separates drop-offs from pick-ups. Objects only appear as dropped or picked. People with broken trajectories can appear in both vertical sections

A possible explanation is specified by a union of events of each type:

$$e = C_{pkdp} \cup C_{dp} \cup C_{pk} \cup C_{none} \tag{1}$$

subject to the constraint that *each person is involved in exactly one event*. This is based on the assumption that each person does not drop/pick more than one object simultaneously. C_{pkdp} is the set of all (i, j, k, l) combinations that define the $pkdp$ events, and similarly for C_{dp}, C_{pk} and C_{none}.

An optimal solution e^* to the assignment problem is obtained by minimizing a cost function. This cost function is defined as a sum of the costs associated with each of the four event types:

$$\begin{aligned} f(e) &= \sum_{C_{pkdp}} f_{pkdp}(p_i, o_j, p_k, o_l) + \sum_{C_{dp}} f_{dp}(p_i, o_j) + \sum_{C_{pk}} f_{pk}(p_i, o_j) + \sum_{C_{none}} f_{none}(p_i) \\ f_{pkdp} &= d(p_i, o_j) + d(o_j, o_l) + d(p_k, o_l \mid o_j) \\ f_{dp} &= f_{pk} = d(p_i, o_i) + \alpha \\ f_{none} &= \beta \end{aligned} \tag{2}$$

where α is a penalty term on unconnected drop-off or pick-up events, and β is a penalty term on a 'none' event.

$d(p_i, o_j)$ is the plausibility of a link between a person and an object, and is derived from the maximum degree of overlap between the bounding box of the object and the bounding boxes of the person across the whole trajectory. If the trajectory of the person is complete, the dropping individuals (identified through area differencing - Section 3) are only connected to dropped objects, and the picking individuals are only connected to picked objects.

$$\begin{aligned} d(p_i, o_j) &= \begin{cases} 1 - \max\left(\frac{sharedBoundingBox(p_i, o_j)}{minBoundingBox(p_i, o_j)}\right) & \text{if } (I(p_i) \subset I(o_j)) \\ \infty & otherwise \end{cases} \\ I(p_i) &= [\text{time entering area}, \text{time exiting area}] \\ I(o_i) &= [\text{time of 'before' reference image}, \text{time of 'after' reference image}] \end{aligned} \tag{3}$$

$d(p_k, o_l \mid o_j)$ is an updated post-segmented cost. When two or more inseparable objects are added, one combined object blob is detected (Section 3). When one of these objects

is subsequently removed, a better estimate of the object's pixels and bounding box can be obtained as Figure 4 shows.

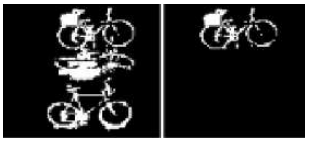


Figure 4: The left image shows three objects dropped simultaneously. As one object is collected (right image), post-segmentation could be achieved

$d(o_i, o_j)$ is the match of picked to dropped objects, and is assessed by comparing corresponding pixels. This match function accommodates any object type, and assumes objects do not change their shape or position between being dropped and picked. If $S(o_i)$ is the set of pixels representing object o_i, then

$$d(o_i, o_j) = \begin{cases} 1 - \frac{|S(o_i) \cap S(o_j)|}{\min(|S(o_i)|, |S(o_j)|)} & \text{if } o_i \in \text{picked} \wedge o_j \in \text{dropped} \wedge I(o_i) > I(o_j) \\ \infty & otherwise \end{cases} \quad (4)$$

4.2 Solving the constrained optimisation problem

To solve the constrained optimisation problem, we propagate a tree of multiple hypotheses (explanations) starting from the beginning of the observation period, and working through to the end, with levels of the tree corresponding to periods of activity. The tree is pruned at each stage to keep the search tractable (beam search) by retaining only the best hypotheses.

As several people enter the storage area simultaneously, several blobs get deposited or picked up. Multiple sets of assignments can be generated to explain the events during one period of activity, and decide who dropped/picked which object. If a person is connected to a picked object, then all matching drop offs that have not been picked up yet are compared for plausibility. The previously unmatched drop event is now joined with its pick up, and can not be picked again.

Each level in the tree is thus expanded into nodes representing the different hypotheses explaining the observations up to the current period of activity. Figure 5 shows a three-level multi-hypotheses tree. Each path (from root to leaf) in the tree corresponds to an explanation. The cost of the path equals the sum of the individual costs of events along that path (except dp events that are superseded by $pkdp$ events). The best path is determined by the minimum cost.

Due to the ambiguities in the visual data, the current best path may not be part of the best path to lower levels of the tree as it propagates into the future. Yet it would be impractical to maintain the complete tree, due to the number of possible hypotheses for all but the simplest cases. If l is the average number of sets of assignments per period of activity, and n is the number of such periods (levels of the tree), then the complexity will be $\Theta(l^n)$. For the first experiment which extended for one hour (refer to Section 5.1 for details), $l = 8.02$ and $n = 35$. This results in an intractable number of leaf hypotheses 4.43×10^{31} of which many are hypotheses of high cost. To ensure scalability over long video sequences involving many periods of activity, only the k-best hypotheses are

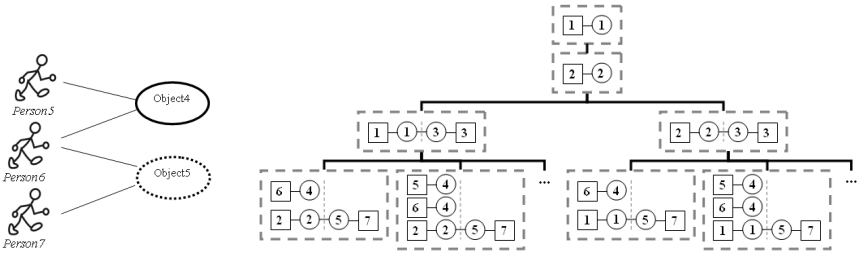


Figure 5: The left graph shows probable connections within a period of activity. Object4 represents a dropped object, while object5 represents a picked object. The various plausible explanations extend the tree. The event $pkdp(p_7, o_5, p_1, o_1)$ can not be added to the left branch as it contradicts the hypothesis at the parent level

retained at each level ($k = 10$ in our experiments). This reduces the complexity to $O(n)$. This method is 'on-line' in that the optimal hypothesis up to the current timestamp is always available.

For comparison purposes, we have implemented both an unconstrained solution (i.e. optimizing $f(e)$ free of the constraint) and one that only partially satisfies the constraint by transforming the problem into a one-to-one assignment problem between individuals and event types. This would still allow dropping people in the *pkdp* events to be involved in another event. This approach is solved using the Munkres (Hungarian) algorithm. The cost matrix input to the assignment problem is an $n \times 4n$ matrix A where each person (row) can be mapped to another person or to one of the other three types of events explained above (columns). If one person can be matched to another person via more than one route, the minimum is selected, as Equation 5 shows.

$$A(i,k) = \min_{j,l} \{ f_{pkdp}(p_i, o_j, p_k, o_l) \} \tag{5}$$

5 Experiments and Results

5.1 Bicycles as Objects

The conducted experiments utilized bicycles as objects and bicycle racks as the storage area. One challenge in using bicycles is the overlap in their parking positions and the difficulty in distinguishing bicycles from each other. Three experiments were conducted. The first experiment extended over 1 hour and staged 28 different cyclists locking their bicycles onto the racks and picking them up again. A view from above was selected to avoid blocking the camera by passing pedestrians or vehicles (Figure 6). Cyclists were aware of the objectives of the research, but not of the techniques used in the solution. The second experiment was staged for 50 minutes and included 9 cyclists. The third full day (9.5 hours) experiment recorded 22 participants using the racks over the course of the day. In all three experiments, simulated thefts were performed. The number of staged thefts was 7, 1 and 5 respectively. Ground truth for the three datasets was established by hand, specifying the correct links between people and bicycles.

5.2 Association Results

The cost function (Equation 2) contains two parameters: α, β. The performance of the constrained solution is relatively insensitive to the value of these parameters, although the partially-constrained and unconstrained solutions are more sensitive to these values. We chose $\alpha = 1.5$ and $\beta = 2.0$ in the experiments presented here. Table 1 shows the percentage of people connected to the correct event in comparison to the ground truth. The constrained solution produced a significant improvement over the unconstrained or partially-constrained solutions. As expected, applying the constraint correctly connects a higher percentage of trajectories, due to the enforced uniqueness.

%	unconstrained	partially-constrained	constrained
exp1	75.86	86.21	93.10
exp2	70.37	70.37	92.59
exp3	83.59	82.03	96.09

Table 1: Percentage of correct connections

An example of an ambiguity that is resolved in the constrained solution from experiment 1 is shown in Figure 7 and Table 2. The unconstrained approach allowed p_{5387} to be linked as the dropping person twice, also p_{187} is involved in two events. The partially-constrained solution solved the problem for p_{5387}, but p_{187} is still involved in two events. The constrained solution satisfied the constraint fully producing the correct connections. Several similar cases across the three experiments could be found.

unconstrained	partially-constrained	constrained
none(187)	none(187)	
pkdp(2916, 3, 187, 1)	pk(2916, 3)	pkdp(2916, 3, 187, 1)
pkdp(8059, 15, 5387, 10)	pkdp(8059, 15, 5387, 10)	pkdp(8059, 15, 5387, 10)
pkdp(8215, 15, 5387, 10)	pkdp(8215, 15, 187, 1)	none(8215)

Table 2: Example from experiment 1 showing how adding constraints improves the associations. The notation for events is defined in Section 4

Figure 6: Viewpoint of the bicycle rack

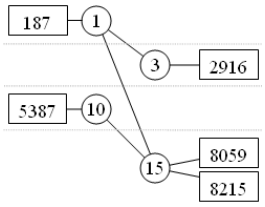
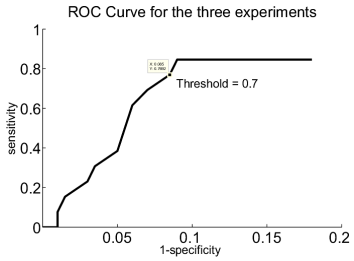


Figure 7: An example from experiment 1. Only the edges required for the example are included

5.3 Colour Comparison and Theft Detection

As an application of the method, we attempted to detect thefts of bicycles by comparing associated individuals using colour information. The presented results use the constrained solution. An $8 \times 8 \times 8$ scale-normalized equal-bin-size RGB colour histogram was generated from the foreground pixels at each frame. 'Scale-by-max' per channel was used as a simple colour constancy algorithm [1]. A per-bin median histogram was calculated across all frames as explained by Bowden and KaewTraKulPong [3]. A distance metric between histograms was produced using histogram intersection [18].



	Predicted	
Actual	Thief	Non-Thief
Thief	10	3
Non-Thief	17	183

Figure 8: ROC curve (left) representing theft detection results. 0.7 was selected as the threshold to calculate the confusion matrix (right) for the three experiments

The ROC curve is shown in Figure 8. At a threshold of 0.7, 77% (10 out of 13) of the theft cases were caught for an 8.5% false-positive rate. 4 false-positive cases resulted from the owner returning wearing different clothing, demonstrating the limitation of using only colour profiles. 4 other false-positive cases were incorrectly connected, while 9 were correctly connected but the colour comparison failed to match the individuals due to poor segmentation from the background.

6 Conclusions and Future Work

The paper has proposed a method for associating individuals as they drop objects off and pick them up sometime later using an online constraint-based optimisation. Ambiguities in the observations are expressed as multiple hypotheses, which can then be verified or invalidated by future observations. Experiments proved the value of the association framework for bicycle theft detection, where colour profiles were used to compare linked individuals. To strengthen the system, the person's features beyond colour could be added to compare individuals, for example, height [4], body mass, gait [13] and behaviour analysis.

Though our experiments were confined to bicycles and bicycle racks, the approach could be applied in other contexts. Car parks, cloakrooms and other parking environments exhibit analogous events and constraints and might also be viable. Although the techniques used to track and detect objects vary across domains, the propagation of multiple hypotheses used to solve the constrained optimisation should remain applicable.

References

[1] K. Barnard, V. Cardei, and B. Funt. A comparison of computational color constancy algorithms—part 1: Methodology and experiments with synthesized data. *IEEE Trans. Image*

Processing, 11(9):972–983, 2002.

[2] J. Berclaz, F. Fleuret, and P. Fua. Robust people tracking with global trajectory optimization. In *Proc. CVPR*, volume 1, pages 744–750, 2006.

[3] R. Bowden and P. KaewTraKulPong. Towards automated wide area visual surveillance: tracking objects between spatially-separated, uncalibrated views. *Proc. Vision, Image and Signal Processing*, 152(2):213–223, 2005.

[4] A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. In *Proc. ICCV*, volume 1, pages 434–441, 1999.

[5] J. Ferryman, editor. *Ninth IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance (PETS2006)*. IEEE, New York, 2006.

[6] N. Gheissari, T. B. Sebastian, and R. Hartley. Person reidentification using spatiotemporal appearance. In *Proc. CVPR*, volume 2, pages 1528–1535, 2006.

[7] A. Gilbert and R. Bowden. Tracking objects across cameras by incrementally learning inter-camera colour calibration and patterns of activity. In *Proc. ECCV*, pages 125–136, 2006.

[8] W. Hu, M. Hu, X. Zhou, T. Tan, J. Lou, and S. Maybank. Principal axis-based correspondence between multiple cameras for people tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(4):663–671, 2006.

[9] O. Javed, K. Shafique, and M. Shah. Appearance modeling for tracking in multiple non-overlapping cameras. In *Proc. CVPR*, volume 2, pages 26–33, 2005.

[10] V. Kettnaker and R. Zabih. Bayesian multi-camera surveillance. In *Proc. CVPR*, volume 2, pages 259 – 265, 1999.

[11] D. Magee. Tracking multiple vehicles using foreground, background and motion models. In *Proc. ECCV Workshop on Statistical Methods in Video Processing*, pages 7–12, 2002.

[12] D. Makris, T. Ellis, and J. Black. Bridging the gaps between cameras. In *Proc. CVPR*, volume 2, pages 205–210, 2004.

[13] M. Nixon and J. Carter. Automatic recognition by gait. *Proceedings of the IEEE*, 94(11):2013–2024, 2006.

[14] M. Piccardi and E. D. Cheng. Track matching over disjoint camera views based on an incremental major color spectrum histogram. In *IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 147–152, 2005.

[15] D. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, 1979.

[16] J. Sivic, C. L. Zitnick, and R. Szeliski. Finding people in repeated shots of the same scene. In *Proc. BMVC*, volume 3, pages 909–918, 2006.

[17] C. Stauffer. Learning to track objects through unobserved regions. In *IEEE Workshop on Motion and Video Computing*, volume 2, pages 96–102, 2005.

[18] M. Swain and D. Ballard. Color indexing. *Int. Journal of Computer Vision*, 7(1):11–32, 1991.

[19] Y. Wang, E. Chang, and K. Cheng. A video analysis framework for soft biometry security surveillance. In *Proc. ACM Int. workshop on Video surveillance & sensor networks*, 2005.

[20] G. Wu, A. Rahimi, E. Y. Chang, G. Kingshy, T. Tsai, J. Ankur, and Y. Wang. Identifying color in motion in video sensors. In *Proc. CVPR*, pages 561–569, 2006.

[21] W. Zajdel, Z. Zivkovic, and B. J. A. Krose. Keeping track of humans: Have i seen this person before? In *Proc. ICRA*, pages 2081–2086, 2005.

[22] L. ZhiHua and K. Komiya. Region-wide automatic visual search and pursuit surveillance system of vehicles and people using networked intelligent cameras. In *Proc. Int. Conference on Signal Processing Proceedings*, volume 2, pages 945–8, 2002.

Spatial modelling of multi-layered LiDAR images using reversible jump MCMC

S. Hernandez-Marin[†], A.M. Wallace[†], G.J. Gibson[‡]
[†]Joint Research Institute on Signal and Image Processing, School of EPS
[‡]Maxwell Institute for Mathematical Sciences, School of MACS
Heriot-Watt University, Edinburgh EH14 4AS
{snh3,a.m.wallace,g.j.gibson}@hw.ac.uk

Abstract

3D imaging LiDAR systems have the potential to acquire multi-layered 3D image data; that is rather than store a single depth value at each pixel, it is possible to store the range to more than one surface within the pixel view direction. Multiple returns are possible at a single pixel when imaging through transparent surfaces, for example when acquiring depth images of cars or buildings that have windows, in which case it is possible to record both external and internal structure. Multiple returns are also possible when the pixel field of view encompasses more than one opaque surface. However, to build such multi-layered 3D images, we need to think of new ways of processing the LiDAR data.

In this paper, we present a unified theory of pixel processing for such data. This is based on a reversible jump Markov chain Monte Carlo (RJMCMC) methodology extended to include spatial constraints by a Markov Random Field with a Potts prior model. We consider two distinct proposal distributions, based on spatial mode jumping and spatial birth/death processes respectively. We also include a delayed-rejection step in the RJMCMC algorithm to improve the estimates of the range and reflectance of each surface element. Our methodology is demonstrated on both photon count and burst illumination LiDAR data.

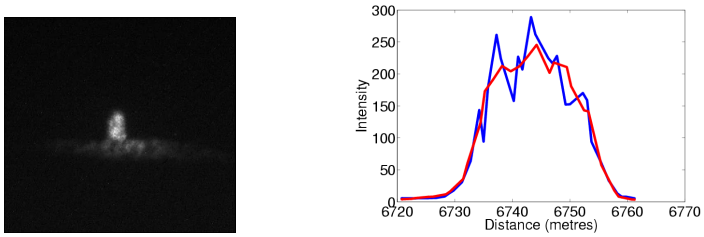
1 Introduction

Our intention is to characterise completely all the 3D surfaces viewed by a Light Detection and Ranging (LiDAR) system. LiDAR works on the principle of time-of-flight, that is the range to the surface can be computed by measuring the go-return time of a laser signal when it impinges on a surface in the field of view (FOV). However, there is no reason why we cannot measure range to more than one surface along the same pixel field of view, if there are transparent surfaces or simply more than one surface in that FOV. Moreover, most scenes contain spatial patterns that have strong dependencies between different pixels. Some pixel configurations are more likely than others since multiple returns can be concentrated in certain regions and completely absent in others. There are many reasons why this occurs, for example, when mapping the pixels to landscape patterns, if a pixel is identified as 'water', it will be most likely surrounded by the same class of pixel. Further, different parts of an object

are related through geometric constraints. If these spatial interactions can be modelled, the classification accuracy can be improved [11].

Many spatial problems are inherently multivariate, in that more than one variable is measured at each spatial location. Multivariate data analysis allows users to display many different spatial data layers. Mardia [8] introduced a multivariate Markov random field (MRF) model for image processing although this work received little attention, due primarily to computational difficulties. Current interest in the analysis of multivariate lattice data has been concentrated on remotely sensed data, especially multispectral images [9], for target detection, scene classification and segmentation.

Often, this multivariate data arises from a time series of measurements. Burst Illumination Laser (BIL) [1] and Time Correlated Single Photon Counting (TCSPC) [7] are examples of 3D LiDAR techniques that can acquire both depth and reflectance images of objects. Fig. 1 shows an example of a single frame acquired with a BIL system and a histogram of integrated intensities from one pixel of Fig. 1(a). As the timing of the camera shutter is varied on the returned pulse, so the intensity rises and falls depending on the depth of the observed surfaces. Fig. 2 shows a small section of a TCSPC pixel image, in which the multiple returns at each pixel are clearly visible as distinct peaks in the photon histograms. Briefly, each pixel records a multivariate measurement which can be considered as an observed photon or intensity histogram which in turn is considered as a sample of a non-normalized statistical mixture distribution. The returns present come not only from the first surface encountered by the projected laser signal, but also from subsequent surfaces in its path. A simple approach to the problem would be to treat the time series of observations at a given image pixel as independent and identically distributed to that for all other pixels, and to analyse them with an appropriate model to understand the underlying theory of the data points. To model such data one can use a parametric approach [7] making use of appropriate mixture distributions and then use the techniques described therein to obtain estimates of the different parameters. However, such an approach can give suboptimal results because it neglects the correlations between the parameters in neighbouring pixels. The contribution of this paper is to incorporate such spatial constraints in the context of RJMCMC pixel processing.



(a) (b)

Figure 1: (a) Single frame BIL image of a trig. point at distance of 6.6km (b) Variation of pixel intensity on the trig point as a function of distance from a BIL system (blue) and final fit from RJMCMC estimation (red)

Spatial dependencies have been introduced in a mixture distribution to take into account spatial heterogeneity using a Bayesian approach [2, 5]. However, the assumption is that the different observations come from a unique mixture distribution in which the number of com-

ponents and some of the parameters (maybe all) are variable. In these approaches, the spatial dependencies were introduced either through the weights of the mixture distribution [2] or through the different allocation variables [5]. Posterior inference was performed using RJMCMC algorithms. Initially, one may consider applying a multivariate MRF to model the spatial dependencies among the different parameters of the different mixture distributions. However, it is not clear how to use these structures in a variable-dimension setting where the different peaks are continuously created or deleted to better explore the space of solutions.

Hence, we consider a complex scenario where spatial interaction is present. To analyse TCSCP and BIL data, we incorporate spatial constraints in a multi-layered image for the first time. Each pixel is a multivariate record, so we use a mixture distribution for each pixel instead of a "global mixture" [5] that includes all the possible returns. To explore the space of solutions, we apply to single pixels a Bayesian statistical approach based on RJMCMC techniques [10] to assess the number, positions and amplitudes of the returned signals from target surfaces [7]. We also include spatial dependencies between pixels that are based on the number of elements of the mixture through the prior distribution as in [6]. However, the major contribution of this paper is to incorporate two new moves within the RJMCMC algorithm, "spatial mode jumping" and a spatial birth/death process that incorporates spatial information in what is now a mixture of proposal distributions. We also introduce a delayed-rejection step [4] for variable-dimension setting to allow a kind of learning process.

2 Model

Consider a rectangular grid of pixels, labeled $m = 1,2,\ldots,N$. At each pixel, a multivariate measurement $y^m = (y_1, y_2, \ldots, y_t)$ is available. Each multivariate measurement is a histogram of photon counts or intensities in which y_i^m is the value recorded in channel i, $i = 1,\ldots,t$. The exact temporal form of the photon count and the intensity histograms are unknown, as they depend respectively on the detector response and camera shutter (time-gate). To interpret such data we follow initially the same approach as [7]. For the TCSPC data, we have used the parametric form of the expected temporal variation of the photon count distribution, employing four piecewise exponential functions. In this study, we assume that the shape parameters of the returned pulses are fixed and known from an instrumental response. For BIL data, we use a look-up table of an instrumental function to interpret the histogram of intensities. This is acquired from the response of the BIL system to a Lambertian reflecting surface at a similar range to the object of interest.

Since our methodology is general, and can be applied to any pulsed LiDAR system, we use the notation f_{op} when referring to the LiDAR operating model. Furthermore, several signals can be present in the same histogram and these will be observed against a finite background level, B_m, whose expected value is considered as a constant across all the channels. The observed histogram in pixel m, $\mathscr{F}_m(i, k_m, \beta_m, t_{0_m}, B_m)$, can be considered as a sample of a statistical mixture distribution with density

$$\mathscr{F}_m(i, k_m, \beta_m, t_{0_m}, B_m) = \sum_{j=1}^{k_m} \beta_{mj} \cdot f_{op}(i, t_{0_{mj}}) + B_m \tag{1}$$

where k_m is the number of peaks, $\beta_m = (\beta_{m1}, \beta_{m2}, \ldots, \beta_{mk_m})$ is a vector of amplitude factors and $t_{0_m} = (t_{0_{m1}}, t_{0_{m2}}, \ldots, t_{0_{mk_m}})$ is a vector of the time of the peak maxima.

If the time resolution is sufficiently fine, the recorded value y_i^m can be considered to be a random sample of a Poisson distribution with intensity equal to $\mathscr{F}_m(i,k_m,\beta_m,t_{0_m},B_m)$, which depends on the different model parameters.

$$y_i^m \sim Poisson(\mathscr{F}_m(i,k_m,\beta_m,t_{0_m},B_m)) \tag{2}$$

To construct a likelihood function, we make two further assumptions. First, the observations in each channel i of the histogram are conditionally independent given the value of the parameters. Second, the spatial dependencies are included in Bayes' equation through parameters representing spatial interactions. Hence, the different $Y = (y^1, y^2, \ldots, y^N)$, are spatially independent given the parameters. The likelihood for the total array is expressed by the following equation

$$L(Y|k,\Omega,T,B) = \prod_{m=1}^{N}\prod_{i=1}^{i_{\max}} e^{-\mathscr{F}_m(i,k_m,\beta_m,t_{0_m},B_m)} \frac{\mathscr{F}_m(i,k_m,\beta_m,t_{0_m},B_m)^{y_i^m}}{y_i^m!} \tag{3}$$

where $k = (k_1,\ldots,k_N)$, $\Omega = (\beta_1,\ldots,\beta_N)$, $T = (t_{0_1},\ldots,t_{0_N})$ and $B = (B_1,\ldots,B_N)$.

3 Bayesian Inference

The objective is inference about the parameters of Eq. 3 to obtain accurate estimates of the number of peaks, position, amplitude and background of the returned signals. If we do not have any spatial contextual information then these unknowns are regarded as drawn from independent prior distributions with full joint prior distribution

$$\pi(k_m,\beta,t_0,B) = \left(\frac{1}{k_{max}}\right)\left(\frac{1}{i_{max}}\right)^k f_G(B_m|c,d)\prod_{j=1}^{k} f_G(\beta_{mj}|a,b) \tag{4}$$

where f_G is the probability density function (pdf) of a gamma distribution.

We consider that observations that correspond to nearby locations are more likely to have similar numbers of peaks than observations from locations that are far apart. Therefore, we generalise the prior distribution given by Eq. 4 including a penalty function, the Potts model, that discourages adjacent pixels from having different numbers of peaks. The Potts model is a generalisation of the Ising model explained in [5] in which the random variable is allowed to have more than two different values. The Potts model has been used previously in image processing applications and in disease mapping applications to model allocation variables. We follow the formulation [5] in which k is modeled jointly

$$p_1(k|\psi) \propto e^{\psi\cdot U(k)} \tag{5}$$

where $U(k) = \sum_{m\sim m'} I[k_m = k_{m'}]$ are the number of like-labeled neighbouring pairs in the configuration $k = (k_1,k_2,\ldots,k_N)$. The parameter ψ is nonnegative and controls the amount of "smoothing", that is, $\psi = 0$ corresponds to a priori spatial independence of the number of peaks and as $\psi \to \infty$ we favour patterns where the number of peaks in neighbouring pixels tend to be similar.

The full joint prior distribution incorporating spatial constraints can be modeled as

$$f(k,\Omega,T,B) = p_1(k|\psi) \times \prod_{m=1}^{N} \pi(k_m,\beta_m,t_{0_m},B_m) \tag{6}$$

Using equations 3, 6 and Bayes' theorem the target distribution can be expressed as

$$\pi(k,\Omega,T,B|Y) \propto L(Y|k,\Omega,T,B)\, f(k,\Omega,T,B) \tag{7}$$

since we just have to know the posterior distribution up to a normalizing constant.

Simulation from the joint probability distribution is difficult. Therefore, we update the different parameters using their respective full conditional distributions. We define a Markov random field with a second-order neighbourhood. The Hammersley-Clifford theorem [3] ensures that we can use the full conditional distribution of Eq. 5.

$$p_m(k_m|\ldots) \propto \exp(\psi \sum_{m' \in \delta_m} I[k_m = k_{m'}]) \tag{8}$$

where δ_m denote the neighbours of m. In this way, a rather complex multivariate probability distribution of a MRF can be obtained by successive simulations from the full conditional distributions.

4 Data Analysis

The Bayesian models used in this work are too complex to be amenable to analytical calculations. RJMCMC techniques [5] allow us to infer the number, positions and amplitudes of the returned signals from target surfaces. We allow moves between state spaces with different dimensionality, which in our case corresponds to changing the number of viewed surfaces. When the dimension is fixed, parameter moves improve the parameter estimates, corresponding to surface range (time of arrival), reflectance (amplitude) and background level. RJMCMC is our preferred method for updating beliefs in response to new information and incorporates prior knowledge. RJMCMC techniques allow us to explore the full posterior distribution of the parameters of the mixture distribution of Eq. 1, given the data values Y supplied by the different LiDAR histograms. The parameters are estimated from the values of a Markov chain whose limiting distribution is a target distribution π. π arises from the posterior distribution defined by Eq. 7. We use simpler versions of π involving only full conditional distributions. The Markov chain constructed involves moves of various types. These are: (a) updates to parameters Ω, T, B, (b) the random birth of a peak, (c) death of a peak, (d) the random splitting of a peak into two peaks and (e) merging of two peaks into a single peak. Movements (b), (c), (d) and (e) are governed by a probability explained in [10]. The implementation of these moves is described in [7].

Our approach also includes a delayed-rejection stage in the RJMCMC algorithm as proposed in [4] in order to improve the mixing of the Markov chain, improving the estimates of the different parameters. In the delayed-rejection strategy, if a candidate move is rejected, we make another attempt to move using a second proposal instead of turning to the next transition. There is no restriction on the number of stages used, but we use only two stages to reduce the computational load. Using delayed-rejection, three different acceptance probabilities have to be calculated. Using Green and Mira's notation [4] for general state spaces, to move from a state x to a state xx, we propose to draw a random number u_1 from a known density g_1. The new stage xx is calculated as $xx = h_1^+(x,u_1)$. The reversed move, from xx to x, is performed by drawing u_1' from g_1' and calculating $x = h_1^-(xx,u_1')$ where h_1^+ and h_1^- are deterministic mappings. The acceptance probability of this first stage is given by

$$\alpha_1(x,xx) = min\left\{1, \frac{\pi(xx)g_1'(u_1')}{\pi(x)g_1(u_1)}\left|\frac{\delta(xx,u_1')}{\delta(x,u_1)}\right|\right\} \tag{9}$$

If this candidate move is rejected, a new move to state z is proposed in a similar way to the previous candidate move; this move is accepted or rejected with a probability given by

$$\alpha_2(x,z) = min\left\{1, \frac{\pi(z)\tilde{g}_1(\tilde{u}_1)\tilde{g}_2(\tilde{u}_2)\{1-\alpha_1(z,xx^*)\}}{\pi(x)g_1(u_1)g_2(u_2)\{1-\alpha_1(x,xx)\}}\left|\frac{\delta(z,\tilde{u}_1,\tilde{u}_2)}{\delta(x,u_1,u_2)}\right|\right\} \tag{10}$$

where $z = h_2^+(x,u_1,u_2)$ and $x = h_2^-(z,\tilde{u}_1,\tilde{u}_2)$. The values $\tilde{u}_1,\tilde{u}_2$ and u_2 are drawn from $\tilde{g}_1,\tilde{g}_2$ and g_2 respectively.

To obtain the numerator of the second acceptance probability we need to calculate a third acceptance probability, $\alpha_1(z,xx^*)$, corresponding to a fictional stationary Markov chain started in z which proposes a move to xx at the first stage, rejects it and accepts a second stage move to x. Although this chain is not really implemented, it is needed to ensure the reversibility condition and therefore preserve the stationary distribution. The expression of the acceptance probability of the virtual stage is identical to that of Eq. 9 with an appropriate change of variables.

In practice, in a 3D secene, objects that have the same attributes tend to cluster in space. Hence, groups of pixels with the same number of peaks and similar parameter values are expected to occur together. Spatial interactions for positions are also expected to occur. However, only spatial interactions for the number of peaks are represented in the prior. Furthermore, the Potts model constrains the number of peaks without considering the relative positions and amplitudes of the current pixel with respect to its neighbouring pixels. Since such contextual information is not available in the prior, we may force situations in which the state space is not explored properly. Therefore, we suggest a proposal distribution for position moves which can exploit the fact that we expect data to show correlated positions. Thus, we have incorporated two new moves, called "spatial mode jumping" and a spatial birth/death process, within the RJMCMC algorithm described in [7]. The "spatial mode jumping" move proposes an update to the position of a peak in pixel m so that its position corresponds to a random perturbation of the position of a peak in pixel m' with some probability. The spatial birth/death move penalises the creation or removal of a peak given the information of the neighbourhood of the pixel. In the spatial birth process, the proposed values for the amplitude and the position of the new peak are drawn from two mixture distributions whose elements incorporate spatial information from neighbouring pixels in a similar way to that of "spatial mode jumping". For a death, one of the current peaks is chosen at random to be removed. Furthermore, the birth and death moves incorporate a delayed-rejection step which learns from the previous rejected value and therefore allows some kind of learning. To model these moves, we use a mixture of proposal distributions which takes account of the current values of the actual peak m and the second-order neighbouring pixels (to reduce algorithm complexity) given by the following expression

$$Q(\cdot) = w_m \cdot q_m^1(\cdot) + \sum_{l\in\delta_m} w_l \left(\sum_{t=1}^{k_{max}} \nu_{lt} \cdot q_t^2(\cdot)\right) \tag{11}$$

where q^1 and q^2 are proposal distributions whose expressions depend on the move we are performing, k_{max} is the maximum number of peaks and w and ν are weights that satisfy $\sum_{l=1}^{9} w_l = 1$ and $\sum_{t=1}^{k_{max}} \nu_{lt} = 1$ respectively. These weights define positive probabilities in such a way that at each step one of the proposals is selected according to these probabilities. The first term of Eq. 11 ensures that a standard independent pixel move is going to be proposed when no peaks are available in the neighbourhing pixels. The second term includes spatial information in the proposal distribution.

5 Experimental Results

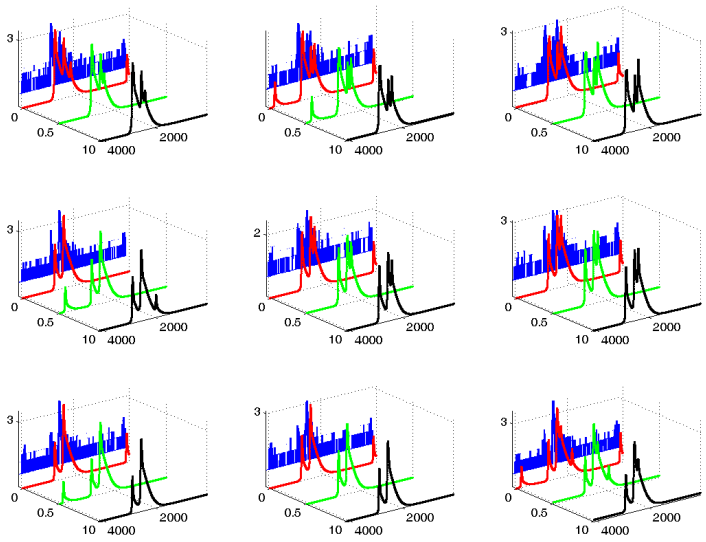
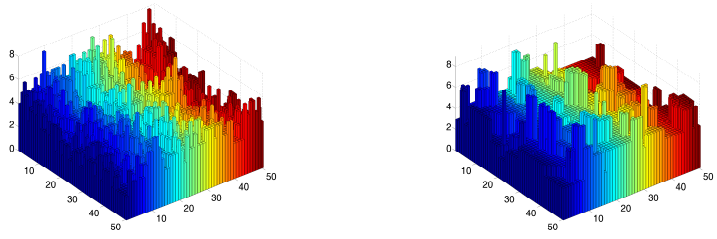


Figure 2: Subimage array of photon count histograms (in blue) of the structure of Fig. 4(a) and final fit without spatial constraints $\psi = 0$ (red), with $\psi = 0.5$ (green) and $\psi = 10$ (black). The vertical axis represent photon counts (in log scale for convenience) whereas the horizontal left axis represents different smoothing parameter values, ψ, and the horizontal right axis represents temporal channels.

In this experiment, we analyse TCSPC and BIL images that correspond to a pixel array of 50 by 50 histograms of photon counts and reflectance/intensity values respectively. The results correspond to 1000 sweeps of the algorithm described in section 4. We infer the number of peaks as that corresponding to the highest marginal posterior probability, $\hat{k} = argmax\ p(k|y)$. Once we determine the number of peaks, we extract estimates of the parameters from $p(\phi|k,y)$ by setting $p(\phi|k=\hat{k},y)$. The values of the parameters are estimated as the mean values of the samples that correspond to such a parameter subspace. The TCSPC histograms come from the structure shown in Fig. 4(a). This data has been analysed with the following values for the parameter $\psi = \{0.5, 5, 10\}$ of the Potts model defined by Eq. 5. The maximum number of peaks in this experiment was set to 10 and the initial number of peaks was selected randomly. Fig. 2 shows an example of subimages of 3 by 3 pixels acquired by TCSPC as well as the final fit obtained with and without spatial constraints. Fig. 3 3(a) and 3(b) display the estimated number when no spatial constraints are considered of peaks and when the smoothing parameter is equal to $\psi = 10$ respectively. As can be seen when using a Potts model the algorithm tends to form clusters, that is, areas with the same number of estimated peaks are connected by paths from neighbour to neighbour.



(a) No spatial constraints (b) $\psi = 10$

Figure 3: Estimated number of peaks of the TCSPC data using RJMCMC

Fig. 4(c) and 4(d) show three layer representations of a scene in which a porcelain cat is placed inside a greenhouse structure. The surface data are represented as point clouds without and with spatial constraints [1] $\psi = 10$. The different surfaces are difficult to perceive with this representation due to the display facilities of Matlab, which was used for code development. However, the shape of the toy cat is discernible and is shown without the other layers in Fig. 4(b). Initially one may think that the number of maximum returns is restricted to 2. Nevertheless, one pixel may have up to 9 returns. The reason of this is due to two main factors. First, our system presents multiple path reflections due to arrangement of the experiment. Second, the beam is impinging on a target with surfaces distributed in depth. Increasing the smoothing parameter enforces corresponding number of surface returns between adjacent pixels; increasing this value further means that the posterior distribution is dominated by the prior information. On the other hand, "false" returns will not be removed if the smoothing parameter is very small. As can be seen from Fig. 4(c) and 4(d), the spatial constraints used "kill" those returns which do not follow any spatial pattern.

The performance of the model is also illustrated on BIL data. The data set chosen is that corresponding to a UK Ordinance Triangulation Point (trig point), imaged at a distance of approximately 6.6*km* from the sensor, shown in Fig. 4(e). As such the LiDAR data is very different from that acquired in a laboratory setting, as the laser is subject to sever turbulence effects. This data presents only one unique return because the surfaces are opaque.. Therefore, only one peak and background returns were considered. The spatial information was incorporated through the proposal distributions and small values of the smoothing parameter $\psi = \{0, 0.5\}$ are considered. The $\psi = \{0.5\}$ value helps to "smooth" the number of peaks in several places where previously it was considered there was only background. If the smoothing parameter is increased, the results obtained are similar to those obtained with $\psi = 0.5$. Fig. 4(f) shows a depth reconstruction of the trig. point.

6 Conclusions

We have described the development and application of RJMCMC techniques incorporating spatial contextual information to process time-of-flight LiDAR data. This spatial information is incorporated on the prior distribution through a Markov Random Field on the number of

[1] Only one spatial-constrained multilayered image is shown due to paper length limitations

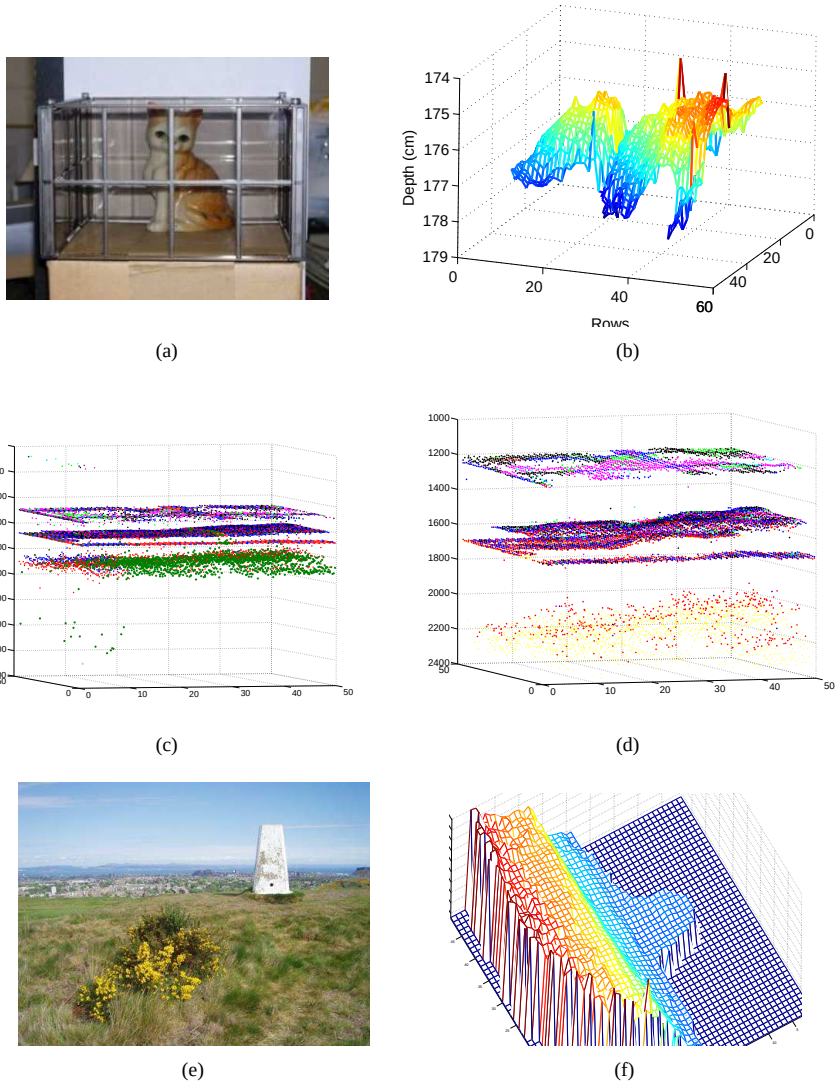


Figure 4: (a) Image of complex multiple return structure composed of a toy cat and a semi-transparent toy greenhouse (b) Mesh of the toy cat of (a) obtained with a TCSPC system (c) and (d) Multiple-layer structure of the experiment (a) with no spatial constraints and smoothing parameter $\psi = 20$ (e) Details from a trig point (distance $\sim$ 6.6km) (f) Depth image obtained with a BIL system using RJMCMC with spatial constraints

peaks, and on the proposal distribution through the development of two new moves: the spatial mode jumping and the spatial birth/death processes. This spatial information can be

used to reduce the influence of clutter of unknown origin by eliminating possible false returns. Further, we have described a delayed-rejection stage which allows us to perform rudimentary learning and therefore improve the performance of this stage.

Our techniques are generally applicable, and have been demonstrated on LiDAR data using both single photon counting and variable temporal gating to extract range measurements. The results are excellent, and show that it is possible to resolve multiple returns and hence characterise objects distributed in 3D space, extracting the underlying spatial structure. For TCSPC LiDAR data, the imaging of multiple surfaces shows a capability well in advance of most 3D LiDAR systems. For BIL data, the detailed and accurate imaging of a the trig point concrete surface and hill grass and earth through a turbulent medium at several Km is also a significant achievement.

References

[1] P. Andersson. Long-range three-dimensional imaging using range-gated laser radar images. *Opt. Eng*, 45(3):1–10, 2006.

[2] C. Fernandez and P. J. Green. Modelling spatially correlated data via mixtures: a Bayesian approach. *J. Roy. Statist. Soc. Ser. B*, 64:805–826, 2002.

[3] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell*, 6:721–741, 1984.

[4] P. J. Green and A. Mira. Delayed rejection in reversible jump Metropolis-Hastings. *Biometrika*, 88:1035–1053, 2001.

[5] P. J. Green and S. Richardson. Hidden Markov models and disease mapping. *J. American Statistical Association*, 97:1055–1070, 2002.

[6] S. Hernandez-Marin, A.M. Wallace, and G.J. Gibson. Creating Multi-layered 3D Images Using Reversible Jump MCMC Algorithms. In *Advances in Visual Computing, LNCS, Springer-Verlag*, pages 405–416, 2006.

[7] S. Hernandez-Marin, A.M. Wallace, and G.J. Gibson. Bayesian analisis of Lidar signals with multiple returns. *IEEE Trans. Pattern Anal. Mach. Intell.*, in press, 2007.

[8] K. V. Mardia. Multi-dimensional multivariate Gaussian Markov random fields with application to image processing. *J. Multivar. Anal.*, 24(2):265–284, 1988.

[9] H. Ren and C. I. Chang. A generalized orthogonal subspace projection approach to unsupervised multispectral image classification. *IEEE Trans. Geoscience and Remote Sensing*, 38(6):2515–2528, 2000.

[10] S. Richardson and P. J. Green. On Bayesian analysis of mixtures with an unknown number of components. *J. Roy. Statist. Soc. Ser. B*, 59:731–792, 1997.

[11] B. Tso and P. M. Mather. *Classification methods for remotely sensed data*. Taylor & Francis New York, 2001.

A New Framework for Automatic Registration of 2D/3D Texture Images

Ayman El-Baz[1] and Georgy Gimel'farb[2]

[1]Bioengineering Department, University of Louisville, Louisville, KY, USA.
[2]Department of Computer Science, University of Auckland, Auckland, New Zealand.

Abstract. A new approach to align an image of a textured object with a given prototype is proposed. Visual appearance of the images, after equalizing their signals, is modeled with a Markov-Gibbs random field with pairwise interaction. Similarity to the prototype is measured by a Gibbs energy of signal co-occurrences in a characteristic subset of pixel pairs derived automatically from the prototype. An object is aligned by an affine transformation maximizing the similarity by using an automatic initialization followed by gradient search. Experiments confirm that our approach aligns complex 2D/3D objects better than popular conventional algorithms.

1 Introduction

Image registration aligns two or more images of similar objects taken at different times, from different viewpoints, and/or by different sensors. The images are geometrically transformed to ensure their close similarity. Registration is a crucial step in many applied image analysis tasks, e.g. to fuse various data sources (such as computer tomography (CT) and MRI data in medical imaging) for image fusion, change detection, or multichannel image restoration; form and classify multi-band images in remote sensing; update maps in cartography, perform automatic quality control in industrial vision, and so forth. Co-registered medical images provide more complete information about the patient, help to monitor tumor growth and verify treatment, and allow for comparing the patient's data to anatomical atlases.

Most of the known registration methods fall into two main categories: feature-based and area-based techniques [1]. Feature based techniques rely on salient local structures extracted from images, e.g. specific areas such as water reservoirs and lakes [2, 3], buildings [4], forests [5], or urban areas [6], specific lines like straight segments [7–9], object contours [10–12], coast lines [13, 14], rivers, or roads [15, 16], and specific points, e.g. road crossings [17], centroids of water areas, or oil and gas pads [18]. Scale invariant feature transform (SIFT) proposed by Lowe [19] is most popular at present because it reliably determines a number of point-wise correspondences between two images differing by affine transformation and local contrast / offset signal deviations. But these methods can be used only if objects have distinctive and non-repetitive local features.

Area-based methods such as the classical least square correlation match directly image signals to avoid feature extraction [20]. However, because the objects are assumed identical to within spatially uniform signal deviations, the correlation is too sensitive to non-uniform and spatially interdependent deviations of the corresponding signals due to sensor noise, illumination variations, and/or different sensor types. Alternative phase correlation and spectral-domain (e.g. Fourier-Mellin transform based) methods [21–25] are more robust with respect to the correlated and frequency dependent noise and non-uniform time varying illumination. However, these methods typically allow for only very limited geometric transformations.

More powerful mutual information (MI) based image registration [26, 27] exploits a probabilistic similarity measure that allows for more general types of signal deviations than correlation. The statistical dependency between two data sets is measured by comparing a joint empirical distribution of the corresponding signals in the two images to the joint distribution of the independent signals. Because the MI-based registration performs the best with multi-modal images [27], it is used in many of medical imaging applications. The joint distribution is estimated using Parzen windows [28, 29] or discrete histograms [30]. The main advantage of the MI is insensitivity to monotone variations of correspondence between the object and prototype signals, but the objects should be of almost identical shape apart from their affine geometrical and monotone signal transformations. The MI allows also for some non-monotone signal correspondence variations although they may change the visual appearance too much and hinder registration accuracy.

We consider a more general case of registering a textured object to a prototype with similar but not necessarily identical visual appearance under their relative 2D/3D affine transformations and monotone variations of signal correspondences. The variations are suppressed by equalizing signals in the images. The co-registered equalized images are described with a characteristic subset of signal co-occurrence statistics. The description implicitly "homogenizes" the images, i.e. considers them as spatially homogeneous patterns with the same statistics. In contrast to the feature-based registration, the statistics characterize the whole object. In contrast to the conventional area-based techniques, similarities between the statistics rather than pixel-to-pixel correspondences are measured. Section 2 represents the equalized object and prototype images as samples of a generic Markov–Gibbs random field (MGRF) with pairwise pixel interaction. Gibbs potentials are analytically estimated from co-occurrence statistics for the prototype. Similarity between an affinely transformed object and the prototype is measured with a total Gibbs energy for a characteristic pixel neighborhood. A new algorithm for selecting the neighborhood for the MGRF model is introduced. After an automatic initialization, the affine transformation aligning the object with the prototype is found by the gradient search for the maximum Gibbs energy of the transformed object. Experiments in Section 3 confirm that our method is more efficient for complex textured objects than more conventional SIFT and MI based registration techniques.

2 MGRF Based Image Registration

Basic notation. We denote $\mathcal{Q} = \{0, \ldots, Q-1\}$; $\mathbf{R} = [(x, y, z) : x = 0, \ldots, X-1; y = 0, \ldots, Y-1; z = 0, \ldots, Z-1]$, and $\mathbf{R}_\mathsf{p} \subset \mathbf{R}$ a finite set of scalar image signals (e.g. gray levels), a 3D arithmetic lattice supporting digital LDCT data $g : \mathbf{R} \to \mathcal{Q}$, and its arbitrary-shaped part occupied by the prototype, respectively. A finite set $\mathcal{N} = \{(\xi_1, \eta_1, \zeta_1), \ldots, (\xi_n, \eta_n, \zeta_n)\}$ of (x, y, z)-coordinate offsets defines neighbors $\{((x+\xi, y+\eta, z+\zeta), (x-\xi, y-\eta, z-\zeta)) : (\xi, \eta, \zeta) \in \mathcal{N}\} \wedge \mathbf{R}_\mathsf{p}$ interacting with each pixel $(x, y, z) \in \mathbf{R}_\mathsf{p}$. The set $\mathcal{N}$ yields a neighborhood graph on $\mathbf{R}_\mathsf{p}$ to specify translation invariant pairwise interactions with n families $\mathcal{C}_{\xi,\eta,\zeta}$ of cliques $\mathcal{C}_{\xi,\eta,\zeta}(x, y, z) = ((x, y, z), (x+\xi, y+\eta, z+\zeta))$ (see Fig. 1). Interaction strengths are given by a vector $\mathbf{V}^\mathsf{T} = \left[\mathbf{V}^\mathsf{T}_{\xi,\eta,\zeta} : (\xi, \eta, \zeta) \in \mathcal{N}\right]$ of potentials $\mathbf{V}^\mathsf{T}_{\xi,\eta,\zeta} = \left[V_{\xi,\eta,\zeta}(q, q') : (q, q') \in \mathcal{Q}^2\right]$ depending on signal co-occurrences; here T indicates transposition.

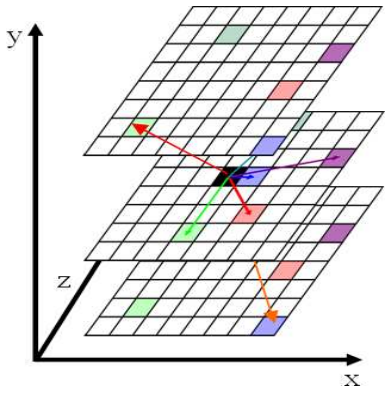


Fig. 1. Pairwise pixel interaction MGRF model.

Image normalization. To account for monotone (order-preserving) changes of signals (e.g. due to different illumination or sensor characteristics), the images are equalized using the cumulative empirical probability distributions of their signals.

MGRF based appearance model. Generic MGRF with multiple pairwise interaction (Fig. 1) [31], the Gibbs probability $P(g) \propto \exp(E(g))$ of an object g aligned with the prototype g° on $\mathbf{R}_\mathsf{p}$ is specified with the Gibbs energy

$$E(g) = |\mathbf{R}_\mathsf{p}|\mathbf{V}^\mathsf{T}\mathbf{F}(g) \tag{1}$$

where $\mathbf{F}^\mathsf{T}(g)$ is the vector of scaled empirical probability distributions of signal co-occurrences over each clique family: $\mathbf{F}^\mathsf{T}(g) = [\rho_{\xi,\eta,\zeta}\mathbf{F}^\mathsf{T}_{\xi,\eta,\zeta}(g) : (\xi, \eta, \zeta) \in$

$\mathcal{N}$] where $\rho_{\xi,\eta,\zeta} = \frac{|\mathcal{C}_{\xi,\eta,\zeta}|}{|\mathbf{R}_\mathsf{p}|}$ is the relative size of the family and $\mathbf{F}_{\xi,\eta,\zeta}(g) = [f_{\xi,\eta,\zeta}(q,q'|g) : (q,q') \in \mathcal{Q}^2]^\mathsf{T}$; here, $f_{\xi,\eta,\zeta}(q,q'|g) = \frac{|\mathcal{C}_{\xi,\eta,\zeta;q,q'}(g)|}{|\mathcal{C}_{\xi,\eta,\zeta}|}$ are empirical probabilities of signal co-occurrences, and $\mathcal{C}_{\xi,\eta,\zeta;q,q'}(g) \subseteq \mathcal{C}_{\xi,\eta,\zeta}$ is a subfamily of the cliques $c_{\xi,\eta,\zeta}(x,y,z)$ supporting the co-occurrence ($g_{x,y,z} = q$, $g_{x+\xi,y+\eta,z+\zeta} = q'$) in g. The co-occurrence distributions and the Gibbs energy for the object are determined over $\mathbf{R}_\mathsf{p}$, i.e. within the prototype boundary after an object is affinely aligned with the prototype. To account for the affine transformation, the initial image is resampled to the back-projected $\mathbf{R}_\mathsf{p}$ by interpolation.

The appearance model consists of the neighborhood $\mathcal{N}$ and the potential $\mathbf{V}$ to be learned from the prototype.

Learning the potentials. The MLE of $\mathbf{V}$ is proportional in the first approximation to the scaled centered empirical co-occurrence distributions for the prototype [31]:

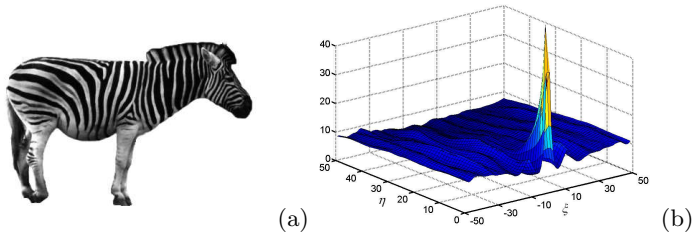
$$\mathbf{V}_{\xi,\eta,\zeta} = \lambda\rho_{\xi,\eta,\zeta}\left(\mathbf{F}_{\xi,\eta,\zeta}(g^\circ) - \frac{1}{Q^2}\mathbf{U}\right);\ (\xi,\eta,\zeta) \in \mathcal{N}$$

where $\mathbf{U}$ is the vector with unit components. The common scaling factor λ is also computed analytically; it is approximately equal to Q^2 if $Q \gg 1$ and $\rho_{\xi,\eta,\zeta} \approx 1$ for all $(\xi,\eta,\zeta) \in \mathcal{N}$. In our case it can be set to $\lambda = 1$ because the registration uses only relative potential values and energies.

Learning the characteristic neighbors. To find the characteristic neighborhood set $\mathcal{N}$, the relative energies $E_{\xi,\eta,\zeta}(g^\circ) = \rho_{\xi,\eta,\zeta}\mathbf{V}^\mathsf{T}_{\xi,\eta,\zeta}\mathbf{F}_{\xi,\eta,\zeta}(g^\circ)$ for the clique families, i.e. the scaled variances of the corresponding empirical co-occurrence distributions, are compared for a large number of possible candidates. For example in 2D case, Fig. 2 shows a zebra prototype and its Gibbs energies $E_{\xi,\eta}(g^\circ)$ for 5000 clique families with the inter-pixel offsets $|\xi| \le 50$; $0 \le \eta \le 50$.

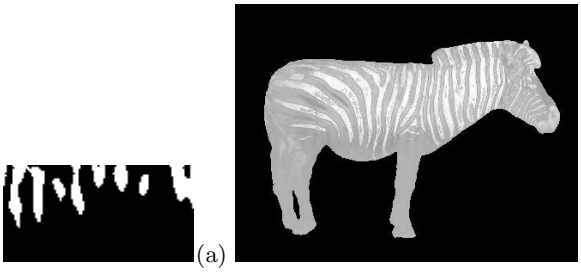
To automatically select the characteristic neighbors, we consider an empirical probability distribution of the energies as a mixture of a large "non-characteristic" low-energy component and a considerably smaller characteristic high-energy component: $P(E) = \pi P_{\mathrm{lo}}(E) + (1-\pi)P_{\mathrm{hi}}(E)$. Both the components $P_{\mathrm{lo}}(E)$, $P_{\mathrm{hi}}(E)$ are of arbitrary shape and thus are approximated with linear combinations of positive and negative discrete Gaussians (efficient EM-based algorithms introduced in [32] are used for both the approximation and estimation of π).

The intersection of the approximate mixture components gives an energy threshold θ for selecting the characteristic neighbors: $\mathcal{N} = \{(\xi,\eta,\zeta) :\ E_{\xi,\eta,\zeta}(g^\circ) \ge \theta\}$ where $P_{\mathrm{hi}}(\theta) \ge P_{\mathrm{lo}}(\theta)\pi/(1-\pi)$. The above 2D example results in the threshold $\theta = 28$ producing 168 characteristic neighbors shown in Fig. 3 together with the corresponding relative pixel-wise energies $e_{x,y}(g^\circ)$ over the prototype:



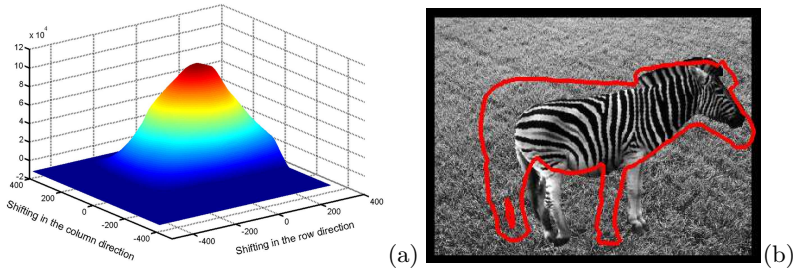
(a) (b)

Fig. 2. Zebra prototype (a) and relative interaction energies (b) for the clique families in function of the offsets (ξ, η).



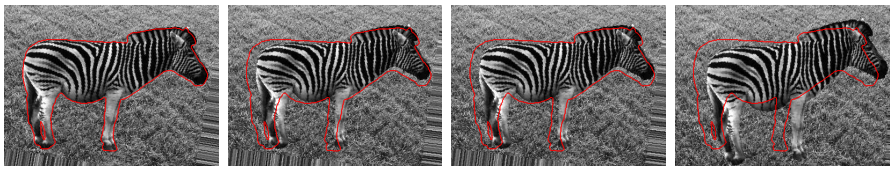
(a) (b)

Fig. 3. (a) Characteristic 168 neighbors among the 5000 candidates (a; in white) and the pixel-wise Gibbs energies (b) for the prototype under the estimated neighborhood.



(a) (b)

Fig. 4. (a) Gibbs energies for translations of the object with respect to the prototype and (b) initial position of the object with respect to the prototype.



(a) (b) (c) (d)

Fig. 5. Our (a), MI-based (b), NMI-based (c), and SIFT-based (d) registration.

$$e_{x,y}(g^\circ) = \sum_{(\xi,\eta)\in\mathcal{N}} V_{\xi,\eta}(g^\circ_{x,y}, g^\circ_{x+\xi,y+\eta})$$

In similar way, the relative energy in the 3D case is calculating as follows:

$$e_{x,y,z}(g^\circ) = \sum_{(\xi,\eta,\zeta)\in\mathcal{N}} V_{\xi,\eta,\zeta}(g^\circ_{x,y,z}, g^\circ_{x+\xi,y+\eta,z+\zeta})$$

Appearance-based registration. The object g is affinely transformed to (locally) maximize its relative energy $E(g_{\mathbf{a}}) = \mathbf{V}^{\mathsf{T}}\mathbf{F}(g_{\mathbf{a}})$ under the learned appearance model $[\mathcal{N}, \mathbf{V}]$. Here, $g_{\mathbf{a}}$ is the part of the object image reduced to $\mathbf{R_p}$ by the affine transformation $\mathbf{a} = [a_{11}, \ldots, a_{33}]$: $x' = a_{11}x + a_{12}y + a_{13}z + a_{14}$; $y' = a_{21}x + a_{22}y + a_{23}z + a_{24}$; $z' = a_{31}x + a_{32}y + a_{33}z + a_{34}$. The initial transformation is a pure translation with $a_{11} = a_{22} = a_{33} = 1$; $a_{12} = a_{13} = a_{21} = a_{23} = a_{31} = a_{32} = 0$, ensuring the most "energetic" overlap between the object and prototype. For example in 2D case, the energy for different translations (a_{14}, a_{24}) of the object relative to the prototype is shown in Fig. 4(a); the chosen initial position (a^*_{14}, a^*_{24}) in Fig. 4(b) maximizes this energy. Then the gradient search for the local energy maximum closest to the initialization selects the six parameters $\mathbf{a}$; Fig. 5 (a) shows the final transformation aligning the prototype contour to the object.

3 Experimental results and conclusions

Due to space limitations, we focus on zebra photos and low dose computed tomography (CT) of human chest commonly perceived as difficult for both the area- and feature-based registration. We compare our approach to three popular conventional techniques, namely, to the area-based registration using MI [27] or normalized MI [30] and to the feature-based registration establishing correspondences between the images with SIFT [19]. Results are shown in Fig. 5.

To clarify why the MI- or NMI-based alignment is less accurate, Fig. 6 compares the MI / NMI and Gibbs energy values for the affine parameters that appear at successive steps of the gradient search for the maximum energy. Both the MI and NMI have many local maxima that potentially hinder the search, whereas the energy is practically unimodal in these experiments. The SIFT-based alignment fails because it cannot establish accurate correspondences between similar zebra stripes (Fig. 7).

In the above example the object aligned with the prototype has mainly different orientation and scale. Figure 8 shows more diverse zebras and their Markov-Gibbs appearance-based and MI-based alignment with the prototype in Fig. 2(a). Visually, the back-projection of the prototype contour onto the objects suggests the better performance of our approach. To quantitatively evaluate the accuracy, masks of the co-aligned objects obtained by manual segmentation are averaged in Fig. 9. The common matching area is notably larger for our approach (91.6%) than for the MI-based registration (70.3%). Similar results obtained for the 3D Lung images are shown in Fig. 10.

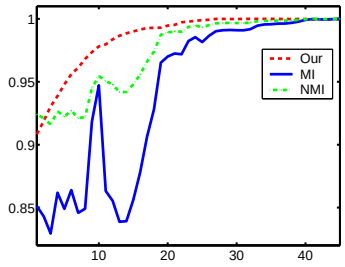


Fig. 6. Gibbs energy, MI, and NMI values at the successive steps of the gradient search.

Fig. 7. Corresponding points by SIFT.

4 Conclusions

In this paper we introduced a new approach to align an image of a textured object with a given prototype whose appearance is modeled with a Markov-Gibbs random field with pairwise interaction. Experimental results confirm that image registration based on our Markov-Gibbs appearance model is more robust and accurate than popular conventional algorithms. Due to the reduced variations between the co-aligned objects, our approach results in more accurate average shape models that are useful, e.g. in image segmentation based on shape priors.

As we mentioned in the experimental result section, the proposed approach is not only limited to zebra photos and lung images but also is suitable for registering starfish photos and brain images. The latter were not included in the paper because of the space limitations, but, the algorithms code, sample data and registration results for the starfishes, brain images will be provided in our web site.

Our future work will focus on integrating our approach in a framework of shape based segmentation; which we believe will enhance the accuracy of segmentation results of the existing approaches.

References

1. B. Zitova and J. Flusser, "Image registration methods: a survey," *Image and Vision Computing*, vol. 21, pp. 977–1000, 2003.

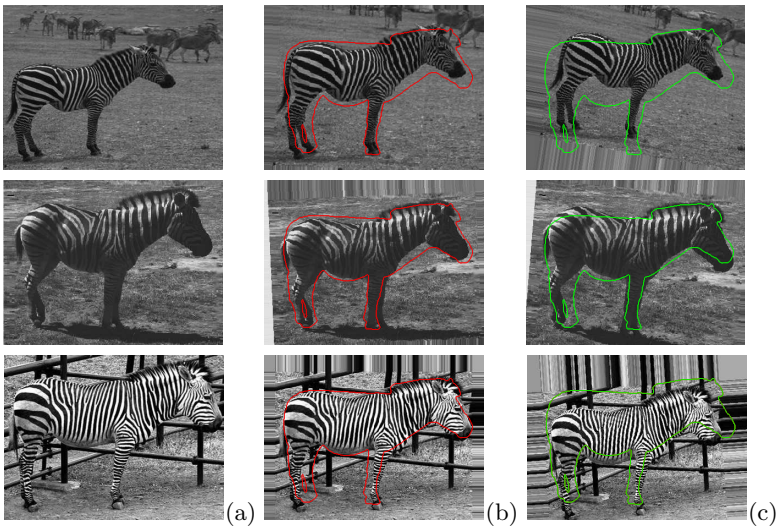


Fig. 8. Original zebras (a) aligned with our (b) and the MI-based (c) approaches.

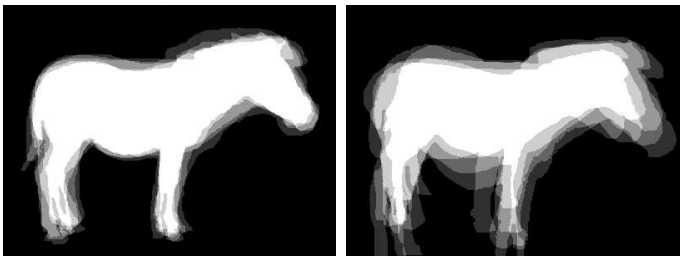


Fig. 9. Overlap between the object masks aligned with our (a; 91.6%) and the MI-based approaches (b; 70.3%).

2. A. Goshtasby and G. C. Stockman, "Point pattern matching using convex hull edges," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 15, pp. 631–637, 1985.
3. M. Holm, "Towards automatic rectification of satellite images using feature based matching," *Proc. Int. Geoscience and Remote Sensing Symp. IGARSS'91*, Espoo, Finland, 1991, pp. 2439–2442.
4. J. W. Hsieh, H. Y. M. Liao, K. C. Fan, M. T. Ko, and Y. P. Hung, "Image registration using a new edge-based approach," *Computer Vision and Image Understanding*, vol. 67, pp. 112–130, 1997.
5. M. Sester, H. Hild, and D. Fritsch, "Definition of ground control features for image registration using GIS data," *Proc. Symp. on Object Recognition and Scene Classification from Multispectral and Multisensor Pixels*, CD-ROM, Columbus, Ohio, 1998.
6. M. Roux, "Automatic registration of SPOT images and digitized maps," *Proc. IEEE Int. Conf. on Image Processing ICIP'96*, Lausanne, Switzerland, 1996, pp. 625–628.

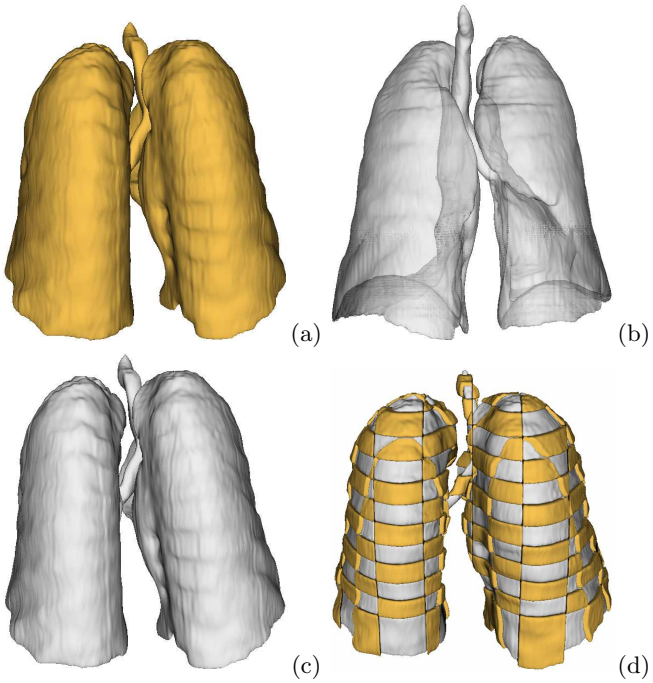


Fig. 10. (a) Reference data, (b) target data, (c) transformed data using 12 degree of freedom affine transformation, and (d) checkerboard visualization to show the quality of rigid alignment. Note that it can be seen that the connectivity at the edges of the lung, between the two volumes is not smooth when using the global model only, this is due to the local deformation which comes from breathing and heart beats.

7. Y. C. Hsieh, D. M. McKeown, and F. P. Perlant, "Performance evaluation of scene registration and stereo matching for cartographic feature extraction," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, pp. 214–237, 1992.
8. S. Moss and E. R. Hancock, "Multiple line-template matching with EM algorithm," *Pattern Recognition Letters*, vol. 18, pp. 1283–1292, 1997.
9. W. H. Wang and Y. C. Chen, "Image registration by control points pairing using the invariant properties of line segments," *Pattern Recognition Letters*, vol. 18, pp. 269–281, 1997.
10. X. Dai and S. Khorram, "Development of a feature-based approach to automated image registration for multitemporal and multisensor remotely sensed imagery," *Proc. Int. Geoscience and Remote Sensing Symp. IGARSS'97*, Singapore, 1997, pp. 243-245.
11. V. Govindu, C. Shekhar, and R. Chellapa, "Using geometric properties for correspondenceless image alignment," *Proc. Int. Conf. on Pattern Recognition (ICPR'98)*, Brisbane, Australia, 1998, pp. 37–41.
12. H. Li, B. S. Manjunath, and S. K. Mitra, "A contour-based approach to multisensor image registration," *IEEE Trans. Image Processing*, vol. 4, pp. 320-334, 1995.
13. H. Maitre and Y. Wu, "Improving dynamic programming to solve image registration," *Pattern Recognition*, vol. 20, pp. 443–462, 1987.

14. D. Shin, J. K. Pollard, and J. P. Muller, "Accurate geometric correction of ATSR images," *IEEE Trans. Geoscience and Remote Sensing*, vol. 35, pp. 997–1006, 1997.
15. S. Z. Li, J. Kittler, and M. Petrou, "Matching and recognition of road networks from aerial images," *Proc. 2*nd *European Conf. on Computer Vision ECCV'92*, St Margherita, Italy, 1992, pp. 857–861.
16. E. H. Mendoza, J. R. Santos, A. N. C. S. Rosa, and N. C. Silva, "Land Use/land Cover Mapping in Brazilian Amazon Using Neural Network with Aster/terra Data," *Proc. Geo-Imagery Bridging Continents*, Istanbul, Turke,y, 2004, pp. 123–126.
17. S. Growe and R. Tonjes, "A knowledge based approach to automatic image registration," *Proc. IEEE Int. Conf. on Image Processing ICIP'97*, Santa Barbara, California, 1997, pp. 228–231.
18. J. Ton and A. K. Jain, "Registering landsat images by point matching," *IEEE Trans. Geoscience and Remote Sensing*, vol. 27, pp. 642-651, 1989.
19. D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. of Computer Vision*, vol. 60, pp. 91-110, 2004.
20. Pope and J. Theiler, "Automated Image Registration (AIR) of MTI Imagery," *Proc. SPIE 5093*, vol. 27, pp. 294-300, 2003.
21. P. E. Anuta, "Spatial registration of multispectral and multitemporal digital imagery using Fast Fourier Transform," *IEEE Trans. Geoscience Electronics*, vol. 8, pp. 353-368, 1970.
22. Q. Chen, M. Defrise, and F. Deconinck, "Symmetric phase-only matched filtering of Fourier-Mellin transform for image registration and recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, pp. 1156-1168, 1994.
23. B. S. Reddy and B. N. Chatterji, "An FFT-based technique for translation, rotation and scale-invariant image registration," *IEEE Trans. Image Processing*, vol. 5, pp. 1266-1271, 1996.
24. G. Wolberg and S. Zokai, "Robust image registration using log-polar transform," *Proc. IEEE Int. Conf. on Image Processing*, Canada, September, 2000.
25. H. Foroosh, J. B. Zerubia, and M. Berthod, "Extension of phase correlation to subpixel registration," *IEEE Trans. Image Processing*, vol. 11, pp. 188-200, 2002.
26. J. Inglada and A. Giros, "On the Possibility of Automatic Multisensor Image Registration," *IEEE Trans. Geoscience and Remote Sensing*, vol. 42, no. 10, October 2004.
27. P. Viola, "Alignment by Maximization of Mutual Information," *Ph.D. dissertation, MIT*, Cambridge, MA, 1995.
28. P. Thevenaz and M. Unser, "Alignment An efficient mutual information optimizer for multiresolution image registration," *Proc. IEEE Int. Conf. on Image Processing ICIP'98,* Chicago, IL, 1998, pp. 833-837.
29. P. Thevenaz and M. Unser, "A pyramid approach to sub-pixel image fusion based on mutual information," *Proc. IEEE Int. Conf. on Image Processing ICIP'96*, Lausanne, Switzerland, 1996, pp. 265–268.
30. C. Studholme, D. L. G. Hill, and D. J. Hawkes, "An overlap invariant entropy measure of 3D medical image alignment," *Pattern Recognition*, vol. 32, pp. 71–86, 1999.
31. G. Gimelfarb and A. A. Farag, "Texture Analysis by Accurate Identification of Simple Markov Models," *Ukrainian Journal of Cybernetics and Systems Analysis*, vol. 41, no. 1, pp. 37–49 January 2005.
32. G. Gimelfarb, A.A. Farag and A. El-Baz, "Expectation-Maximization for a linear combination of Gaussians," *Proc. of IAPR International Conference on Pattern Recognition, ICPR-2004, Cambridge, UK*, August 2004, pp. 422425.

Fast Multigrid Optimal Mass Transport for Image Registration and Morphing

Tauseef ur Rehman, Gallagher Pryor and Allen Tannenbaum
Georgia Institute of Technology, Atlanta GA, USA

Abstract

In this paper we present a novel, computationally efficient algorithm for non-rigid 2D image registration based on the work of Haker *et al.*[1, 2]. We formulate the registration task as an Optimal Mass Transport (OMT) problem based on the Monge-Kantorovich theory. This approach gives a number of advantages over other conventional registration methods: **(1)** It is parameter free and no landmarks need to be specified, **(2)** it is symmetrical and the energy functional has a unique minimiser, and **(3)** it can register images where brightness constancy is an invalid assumption. Our algorithm solves the Optimal Mass Transport program via multi-resolution, multi-grid, and parallel methodologies on a consumer graphics processing unit (GPU). Although solving the OMT problem has been shown to be computationally expensive in the past, we show that our approach is almost two orders magnitude faster than previous work and is capable of finding transport maps with optimality measures (mean curl) previously unattainable by other works (which directly influences the quality of registration). We give results where the algorithm was used to register 2D short axis cardiac MRI images and to morph two image sets from a SOHO solar flare image sequence.

1 Introduction

1.1 Image Registration and Morphing

Image registration and morphing are amongst the most common image processing problems. Registration is necessary in order to compare or integrate image data obtained from different measurements while image morphing, on the other hand, is a class of techniques that deals with the metamorphosis of one image into another (also known as image interpolation). Given two related images, these two techniques can be used together to generate a sequence of intermediate images in which an image gradually changes into the other over time. Critical to the success of this process is the quality of the warping function generated by the registration procedure. Morphing is achieved by shifting all pixels in the image according to the vector field defined by this warping function and will not be reasonable if the underlying warping function is not correct.

1.2 Optimal Mass Transport

The optimal mass transport problem was first formulated by a French mathematician Gasper Monge in 1781, and was given a modern formulation in the work of Kantorovich [3] and, therefore, is now known as the *Monge-Kantorovich problem*. The original problem concerned finding the optimal way to move a pile of soil from one site to another in the sense of minimal transportation cost. Hence, the Kantorovich-Wasserstein distance is also commonly referred to as the *Earth Mover's Distance (EMD)*.

Recently, Haker et al. have applied the optimal mass transport approach to certain medical image registration problems [1, 2]. Rigorous mathematical details for their algorithm are given by Angenent et al. [4]. Although there have been a number of algorithms in the literature for computing an optimal mass transport, the method by Haker et al. computes the optimal warp from a first order partial differential equation, which is a computational improvement over earlier proposed higher order methods [5, 6] and computationally complex discrete methods based on linear programming. The OMT method has a number of distinguishing characteristics: **(1)** It is parameter free and no landmarks need to be specified, **(2)** it is symmetrical and the energy functional has a unique minimiser, and **(3)** it can register images where brightness constancy is an invalid assumption. Conducting registration via OMT also utilizes all of the greyscale data in both images and places the two images on equal footing. It is thus symmetrical; the optimal mapping from image A to image B is the inverse of the optimal mapping from B to A. Furthermore, OMT does not require that landmarks be specified and the minimizer of the distance functional involved is unique; there are no other local minimisers. Finally, OMT is specifically designed to take into account changes in densities that result from changes in area or volume.

However, solving the Optimal Mass Transport problem is computationally expensive and a typical 2D image registration using the method of [1, 2] has been reported to require a matter of minutes. It has also been shown that this method may not necessarily converge to an optimal solution within a feasible amount of time. In our previous work [16] we showed considerable improvement in computation time when implemented using multi-resolution strategy.

Contribution. In this paper we present a new algorithm based on [1, 2] for solving the Optimal Mass Transport Problem using a *coarse-to-fine* strategy and a *full multigrid solver*, which we implement on a consumer *graphics processing unit*. We demonstrate two orders of magnitude improvement in computation time and that the algorithm is capable of computing transport maps with optimality measures (mean curl) previously unattainable by other works. The latter development has a direct influence on the quality of registration and image warps obtained by the algorithm while the speed-up achieved by our algorithm makes the use of OMT on 3D grids and other large bodies of data feasible and furthers the applicability of OMT for image processing tasks.

2 Optimal Mass Transport Theory

2.1 Formulation of the Problem

We will briefly provide an introduction to modern formulation of the Monge-Kantorovich problem. We assume we are given, *a priori,* two sub-domains Ω_0 and Ω_1 of $\boldsymbol{R}^d$ with smooth boundaries, and a pair of positive density functions, μ_0 and μ_1 defined on Ω_0 and Ω_1 respectively. We assume that,

$$\int_{\Omega_0} \mu_0 = \int_{\Omega_1} \mu_1 \tag{1}$$

This ensures that we have same total mass in both the domains. We now consider *diffeomorphisms* $\tilde{u}$ from Ω_0 to Ω_1 which map one density to other in the sense that,

$$\mu_0 = \left|D\tilde{u}\right| \mu_1 \circ \tilde{u} \tag{2}$$

which we call the *mass preservation* (MP) property, and write $\tilde{u} \in MP$. Equation (2211) is called the *Jacobian equation.* Here, $|D\ \tilde{u}|$ denotes the determinant of the Jacobian map $D\ \tilde{u}$, and $\circ$ denotes composition of functions. It basically implies that if a small region in Ω_0 is mapped to a larger region in Ω_1, then there must be a corresponding decrease in density in order for the mass to be preserved. There may be many such mappings, and we want to pick an optimal one in some sense. Accordingly, we define the squared L^2 Monge-Kantorovich distance as following:

$$d_2^2(\mu_0, \mu_1) = \inf_{\tilde{u} \in MP} \int \left\| \tilde{u}(x) - x \right\|^2 \mu_0(x) dx \tag{3}$$

The *optimal MP map* is a map which minimizes this integral while satisfying the constraint (2). The Monge-Kantorovich functional (3) is seen to place a penalty on the distance the map $\tilde{u}$ moves each bit of material, weighted by the material's mass. A fundamental theoretical result[7-9], is that there is a unique optimal $\tilde{u} \in MP$ transporting μ_0 to μ_1, and that $\tilde{u}$ is characterized as the gradient of a convex function ω, i.e., $\tilde{u} = \nabla \omega$. This theory translates into a practical advantage, since it means that there are no non-global minima to stall our solution process.

2.2 Computing the Transport Map

We will describe here only the algorithm for finding the optimal mapping $\tilde{u}$. The details of this method can be found in [2]. The basic idea for finding the optimal warping function is first to find an initial MP mapping u^0 and update it iteratively to decrease an energy functional. When the pseudo time t goes to ∞, the optimal u will be found, which is $\tilde{u}$. Basically, there are two steps:

2.2.1 Finding an Initial Mapping

The first step in this algorithm is to find an initial mass preserving mapping. This can be done for general domains using the method of Moser [10] or the algorithm proposed in [2]. The later method can simply be interpreted as the solution of a one-dimensional Monge-Kantorovich problem in the x-direction followed by the solution

of a family of one-dimensional Monge-Kantorovich problems in *y*-direction. We used this later method for our implementation and experiments.

2.2.2 Finding the Minimizer

The second step is to adjust the initial mapping found above iteratively using gradient descent in order to minimize the functional defined in Equation (3), while constraining u so that it continues to satisfy Equation (2). This process iteratively removes the curl from the initial mapping u and, thereby, finds the polar factorization of u. For details on this technique, please refer to [2]. The overall algorithm is summarized graphically in Figure (1).

2.3 Defining the Warping Map

In elastic registration applications, one usually wants to visualize the explicit warping between the two images where one image smoothly deforms to the other. This has been shown to be easily done using the solution from the Monge-Kantorovich problem described above using the following relationship [2]:

$$X(x,t) = x + t(\tilde{u}^*(x) - x) \qquad (4)$$

where, $X(x,t)$ defines our continuous warping map between densities μ_0 and μ_1. Note that when $t = 0$, X is the identity map and when $t=1$, it is the solution $\tilde{u}^*$ to the Monge-Kantorovich problem. All the morphs in the results section have been created using this equation.

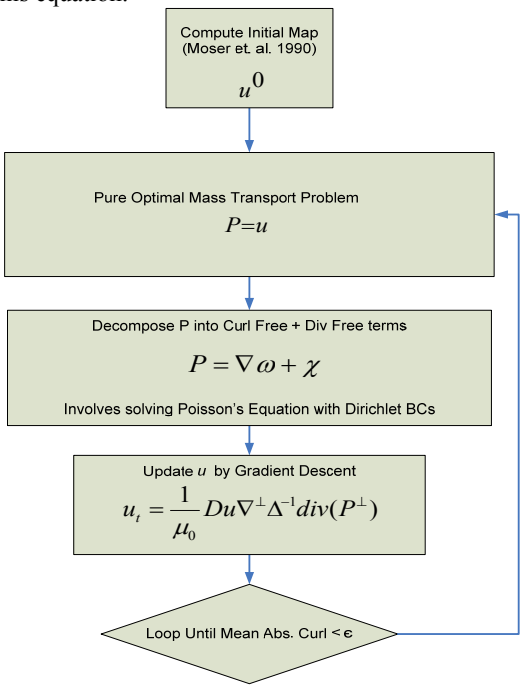


Figure 1: Optimal Mass Transport Algorithm

3 Multi-resolution OMT

Performing image registration using a multi-resolution approach is widely used to improve speed, accuracy and robustness. The basic idea is that registration is first performed at a coarse scale. The spatial mapping determined at the coarse level is then used to initialize registration at the next finer scale. This process is repeated until it reaches the finest scale. This *coarse-to-fine strategy* greatly improves the registration success rate and also increases robustness by eliminating local optima at coarser scales [11]. Our coarse-to-fine hierarchy comprised of three levels and uses bi-cubic interpolation to interpolate the solution from the coarse to fine grids.

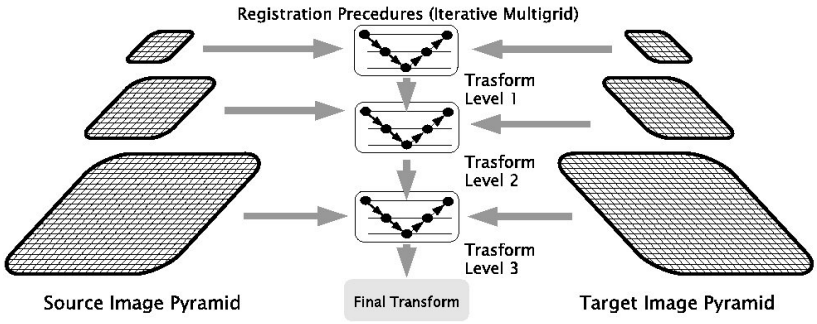


Figure 2: Coarse-to-fine Approach

This multi-resolution technique diminishes the mean curl of u remarkably fast as compared to the normal case as is shown in Figure 3 below. These results correspond to the heart registration example given in Figure 8. Figure 3 shows the mean curl of u achieved in the same number of iterations with and without multi-resolution. It should be noted that same number of iterations of multi-resolution complete in less than 1/3 the time.

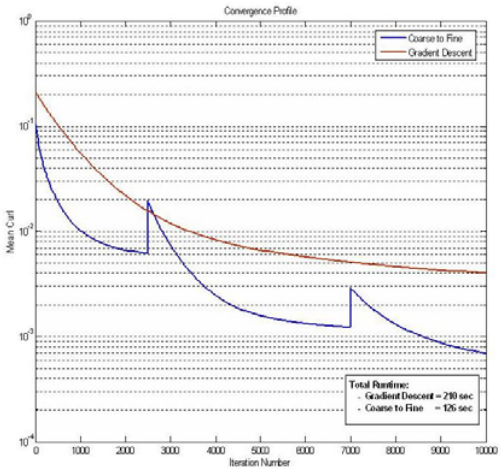


Figure 3: Convergence with and without multi-resolution approach

3.1 Multigrid

We used the *Full Multigrid* (FMG) algorithm in our implementation of the Poisson solver also known as the *nested iteration.* In the case of FMG, the initial approximation is obtained by interpolating from a coarse-grid solution. For discretization we used the standard five-point discrete approximation of the Laplacian given by:

$$\nabla^2 u_{ij} \approx u_{i-1,j} + u_{i+1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j} \qquad (5)$$

where, i and j are row and column indices into the grid. We chose the most frequently used Standard Coarsening, i.e. doubling the mesh size in each direction. The same discretization of the Laplace operator is used on all the grid sizes. This is known as *Discrete Coarse Grid Approximation* (DCA). We *restrict* the residual from the fine-to-coarse transfer using *Half Weighting* technique and for *interpolation* from coarse-to-fine transfers we use *Bilinear Interpolation.* For a comprehensive overview of multigrid methods, we refer to [12, 13].

The FMG algorithm is graphically illustrated in the following figure. The thick arrows indicate use of a higher order interpolation (bi-cubic) scheme instead of the Bilinear Interpolation. This is due to the fact that when we increment to the next to coarsest level, we do not have to use an interpolation complementing the Full weighting scheme to maintain symmetry in a cycle.

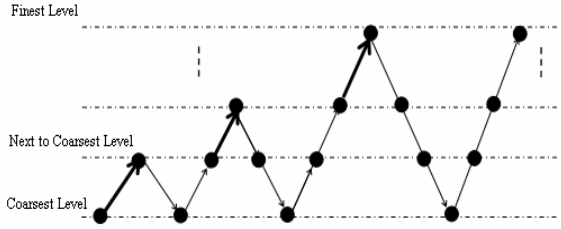


Figure 4: Structure of one Full Multigrid cycle

3.2 GPU Implementation

An advantage of our solution to the OMT problem is that it is particularly suited for implementation on parallel computing architectures making optimization with multiple processing cores highly advantageous. Over the past few years, it has been shown that GPUs are particularly suited for these types of parallelizable problems [14, 15].

Implementing the PDE-based OMT solver on the GPU comes quite naturally as it can be abstracted down to a series of convolutions and simple point-wise arithmetic between data grids. From this standpoint, the GPU has much to gain over the CPU implementation: while the CPU computes updates on data grids one element at a time, the GPU is capable of updating entire grids in one pass due to their massively parallel architecture. This is graphically illustrated in Figure 5 below. Such simplicity allowed us to implement the entire algorithm via an OpenGL/fragment shader approach using the same methodology as [14, 15] for GPU computation. In contrast to [14,15], however, the OMT solver is much more

complex in that a multigrid solver is only a component of the algorithm in addition to additional computation of temporary quantities (curl, divergence, coarse-to-fine interpolation) and an upwinding scheme for state updates.

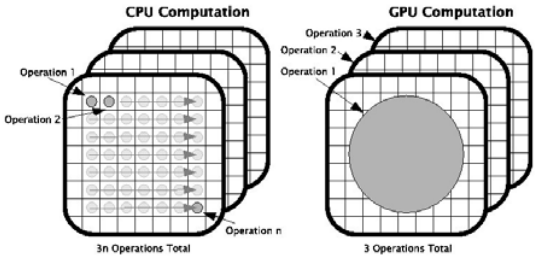


Figure 5: CPU vs. GPU solutions of PDEs.

The GPU implementation of the algorithm gave considerable performance gains over the CPU. For instance, on a 512^2 grid, the time required to perform one iteration of the solver by Haker et. al. was 0.42 seconds while the GPU implementation required 0.0159 seconds, representing an improvement of 2681 percent (Figure 6). This improvement increases with grid size to 7060 percent on a 2048^2 grid. All results were obtained on a Dual Xeon 1.6GHz and nVidia GeForce 7950 GX2 graphics card with a 3DMark score of 6747.

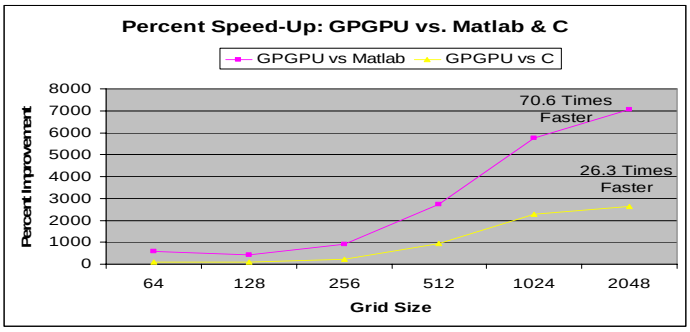


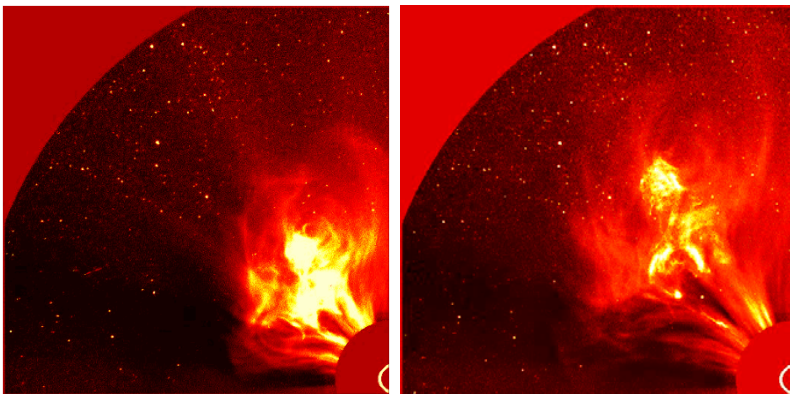
Figure 6: Comparison of execution speeds at each grid size.

4 Examples and Results

We illustrate our method using two examples. In the first case, we interpolate between two greyscale SOHO solar flare images shown in colour here to highlight the details. We take Figure 7(a) and Figure 7(b) to be the starting and ending images, respectively. Figure 8 shows the intermediate images generated by our morphing algorithm at times t=0.25, 0.5 and 0.75. In Figure 9 we show that a classical non-rigid registration based on optical flow fails to warp the flare images. The procedure cannot account for the new information present in the second image versus the first image and thus the warp fails.

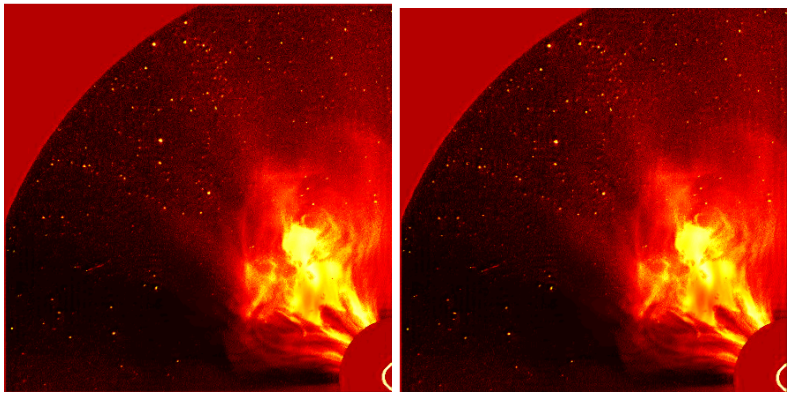
In the second case, we applied our method to two different 2D short axis cardiac MRI images (256^2 pixels) taken from a sequence of images (Figure 10). The images were inverted for improved performance. Note that the deformed grid

(Figure 11) shows areas of compression of the ventricle. Note that in each case, total computation time was only a matter of seconds.

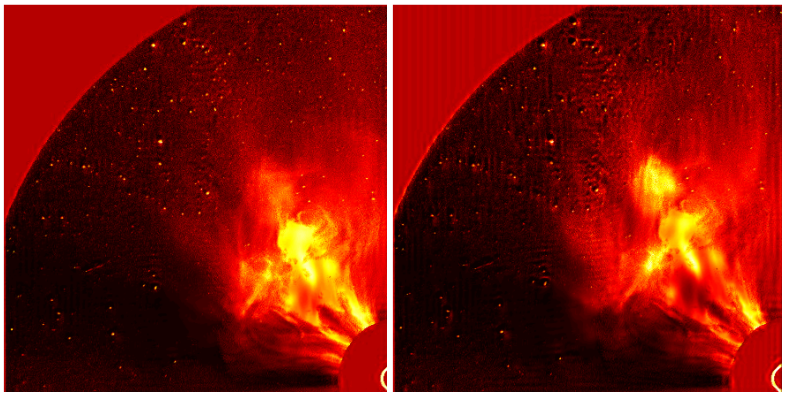


(a) Source Image (b) Target Image

Figure 7: Source and target solar flare images (512^2 grid)



(a) $t = 0.25$ (b) $t = 0.50$



(c) $t = 0.75$ (d) $t = 1.00$

Figure 8: OMT-interpolated solar flare images (12 sec, μ curl $10^{-3.2}$)

110

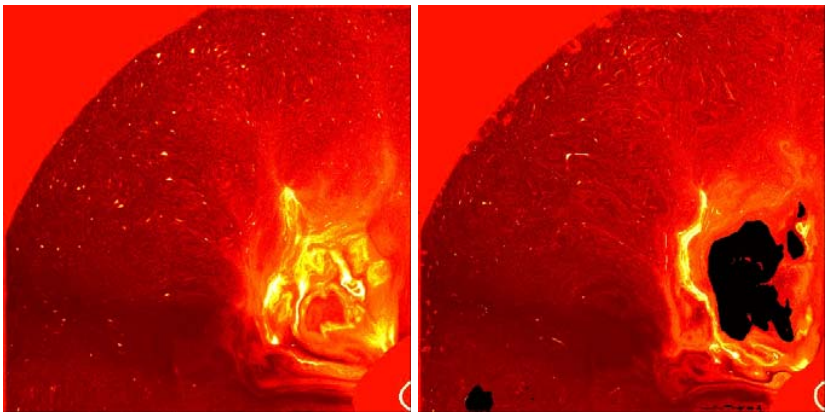


Figure 9: Failure of classical non-rigid registration based on optical flow warp.

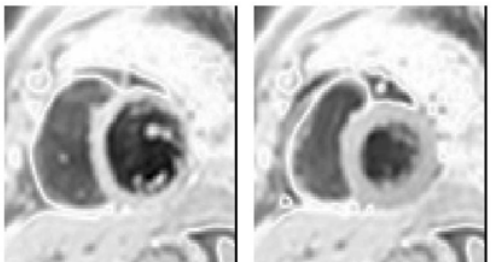


Figure 10: Two frames from 2D cardiac MRI image sequence.

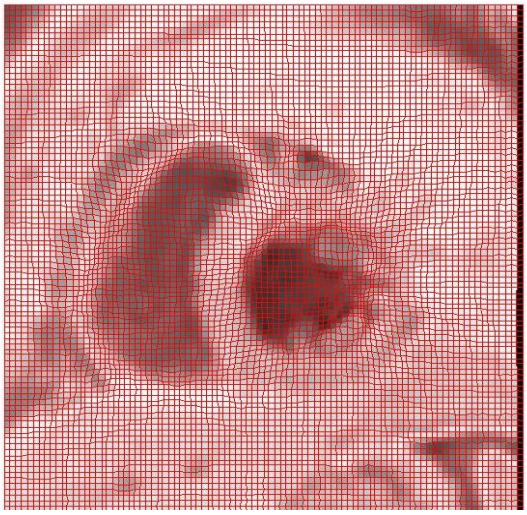


Figure 11: Deformed grid for cardiac example (64 sec, μ curl $10^{-3.6}$).

5 Conclusions

In this paper, we presented a computationally efficient method for image registration and morphing based on the classical problem of optimal mass transportation. The speed-up achieved by our algorithm makes the use of OMT on 3D grids and other large bodies of data feasible and furthers the applicability of OMT for image processing tasks including compression and coding.

References

[1] S. Haker and A. Tannenbaum, "Optimal Mass Transport and Image Registration," in *IEEE Conference on Variational and Level Set methods in Computer Vision*, 2001.

[2] S. Haker, L. Zhu, A. Tannenbaum, and S. Angenent, "Optimal Mass Transport for Registration and Warping," *International Journal of Computer Vision,* vol. 60, pp. 225-240, 2004.

[3] L. V. Kantorovich, "On a problem of Monge," *Uspekhi Mat. Nauk.,* vol. 3, pp. 225-226, 1948.

[4] S. Angenent, S. Haker, and A. Tannenbaum, "Minimizing flows for the Monge-Kantorovich problem," *SIAM Journal of Mathematical Analysis,* vol. 35, pp. 61-97, 2003.

[5] G. E. Christensen, "A deformable neuroanatomy handbook based on viscous fluid mechanics," in *27th Annual Conference on Information Sciences and Systems*, 1993, pp. 211-216.

[6] G. E. Christensen, R. D. Rabbit, and M. Miller, "Deformable templates using large deformation kinetics," *IEEE Transactions on Image Processing,* vol. 5, pp. 1435-1447, 1996.

[7] Y. Brenier, "Polar factorization and monotone rearrangement of vector-valued functions," *Comm. Pure Appl. Math.,* vol. 64, pp. 375-417, 1991.

[8] W. Gangbo and R. McCann, "The geometry of optimal transportation," *Acta Math.,* vol. 177, pp. 113-161, 1996.

[9] M. Knott and C. Smith, "On the optimal mapping of distributions," *J. Optim. Theory,* vol. 43, pp. 39-49, 1984.

[10] J. Moser, "On the volume elements on a manifold," *Transactions of American Mathematics Society,* vol. 120, pp. 286-294, 1965.

[11] T. S. Yoo, *Insight into images, Principles and Practice for Segmentation, Registration and Image Analysis*: A. K. Peters Ltd., 2004.

[12] W. L. Briggs, V. E. Hensen, and S. F. McCormick, *A Multigrid Tutorial*: SIAM, 2000.

[13] U. Trottenberg, C. Oosterelee, and A. Shulller, *Multigrid*: Academic Press, 2001.

[14] J. Bolz and e. al., "Sparse matrix solvers on the GPU: Conjugate Gradients and Multigrid," in *SIGGRAPH*, 2003, pp. 917-924.

[15] G. Nolan and e. al.. "A multigrid solver for boundary value problems using programmable graphics hardware," in *SIGGRAPH*, 2003, pp. 102-111.

[16] T. Rehman and A. Tannenbaum, "Multigrid Optimal Mass Transport for Image Registration and Morphing", in *SPIE Conference on Computational Imaging V*, 2007

Rotation, Rescaling and Occlusion Invariant Object Retrieval

Michela Lecca and Stefano Messelodi
Fondazione Bruno Kessler - IRST
via Sommarive 18, 38050 Povo, Trento, Italy
{lecca, messelod}@itc.it

Abstract

This paper presents a new approach for rotation, rescaling and occlusion invariant retrieval of the objects of a given database D. The objects are represented by means of many 2D views and each of them is occluded by several half-planes. The remaining visible parts (*linear cuts*) as well as the whole views are stored in a new database D' and described by low-level features. Given a portion *R* of an image, the retrieval of the most similar object is done by generating some linear cuts of *R*, and by comparing their descriptors with those of the elements of D'. Some heuristic rules regarding visual similarity and geometric properties of the objects in the database drive this process. In the case *R* is recognized as an object partially occluded, a strategy for the reconstruction of the whole shape of *R* is also presented.
The tests carried out on synthetic and real-world datasets showed good performances both in recognition and in reconstruction accuracy.

1 Introduction

The diffusion of the digital photo- and video- cameras in the daily life, the rapid increase of the visual materials, their simple and cheap availability, the development of efficient storage devices, make the use of automatic image retrieval systems necessary to manage, index and organize the visual documents in easy and fast way [5].

A content-based image retrieval (CBIR) system is a tool for browsing and searching images from a large database. The system input is a digital image (query) and the output is the database elements or a subset of them ranked by similarity with respect to the query. The similarity is defined as distance between some features describing the database images and the query. Examples of CBIR systems are COMPASS [3], Virage [11], QBIC [7], VisualSEEk [17].
The object retrieval systems are CBIR systems for browsing and searching for objects in a large database of digital images. In this case, the query is a region of a digital image and the output is the database objects or a subset of them ordered by similarity respect to the query. Due to their many applications in the e-commerce and in the automatic indexing of digital images, the development of such systems is an attractive topic in Computer Vision and especially in Image Understanding. An example of commercial application is the search for web catalog products visually similar to a given input. Moreover, these retrieval engines can be inserted in systems for the recognition of objects in digital images

to provide an automatic indexing and a high-level description of images [10]. Since the objects can appear in various accidental circumstances that modify their appearance (like perspective transformations, affinities, occlusions or changes of illuminant), the main bottleneck is to find a description of the objects, able to permit a stable recognition under a large number of conditions.

The most popular approaches describe the objects and the image portion by means of *local features*, like pixel intensity, corners, high curvature points, edge fragments [13], [9], [16], [6]. Other methods describe the objects and the image region by some subsets of pixels [12], or by dividing the object image into many regular parts, for example rectangles, and by describing each of them by global features, [8], [4]. For objects consisting of distinguishable parts arranged in a fixed spatial configuration, models composed by a description of the parts along with information about their spatial relationships or even with a label about their functionality are proposed [14]. Generally, in all these models, the local features are organized in hierarchical structures, for instance trees or graphs, and recognition is reduced to their matching [16], [18].

The choice of the local features for the object description, their extraction from the image, and the matching algorithm for the recognition are often computationally expensive, requiring generally an analysis pixel by pixel or in windows centered in each pixel, and even more for finding the feature correspondences, also when a priori knowledge or heuristic rules are introduced [2].

In this work we propose a method for the retrieval of objects invariant with respect to rotation and/or rescaling and/or occlusions. Differently from the most common approaches, our approach does not use local features for the description of the objects and of the image region to be classified, and the computation of the descriptors as well as the features matching are very fast. The objects are modeled by a set of 2D views and by some occlusions of them by half-planes; given an input region R, the retrieval of the most similar object is done by generating some occlusions of R by half-planes and by comparing them with those of the objects. An algorithm for the reconstruction of the whole shape of occluded objects is also proposed. The main advantages in the use of our method are the simplicity of the object model, the invariance by rescaling, rotating and/or occlusions, and the restricted user interaction. The main disadvantage is due to the fact that the method needs a large amount of memory space to store the database containing the objects and their cuts, but a distributed architecture can easily solve this problem.

2 Overview on the Method

Our method consists of three parts detailed in the next Sections: (1) object model construction, (2) recognition algorithm, (3) reconstruction of recognized occluded objects.
In our approach, the objects to be recognized are represented by many 2D views and they are stored in a database D. Moreover each view is occluded by several half-planes having different slope and masking different percentage of the view. A new database D' containing the remaining visible parts (*linear cuts*) and D is then built.
To establish if a region R of an input image is a view possibly occluded of an object of D, a set of linear cuts of R is generated and then compared them with the items of D'. Each element of D', R, and each linear cut of it are described by means of a vector of low-level features, encoding information about color, shape and texture. The pairs (C_R, C_{O_v}), where

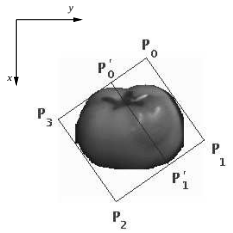


Figure 1: Rectangles for the generation of the linear cuts of a view of an object of the well-known database COIL-100 (`http://www.cs.columbia.edu/CAVE/`).

C_R is a linear cut of R and C_{O_v} is the element of D' having the smallest distance from it (*recognition distance*), are stored in a *candidate list*. R is recognized as an instance of an object O in D if the majority of the linear cuts of R is associated to a linear cut of views of O. Since the descriptors are invariant by rescaling, rotation, and composition of thereof, the system is able to recognize also rotated and/or rescaled partially occluded views. The scale factor of R and its orientation in the image plane with respect to the associated object view are determined and in case of occlusions, they are used for the reconstruction of the whole shape of R. The recognition algorithm is guided by parametric heuristic rules related to geometric properties (area, scale factor) and visual similarity (recognition distance). If R corresponds to an occluded object view, its whole shape is reconstructed and its scale factor and orientation with respect the recognized view are returned. Some thresholds used in the recognition and reconstruction procedures are estimated automatically, while others are user inputs.

Visual similarity between a portion of R and an element of D' is defined as the L^1-distance between their correspondent feature vectors. We call *recognition distance* the minimum distance between R and the items of D'. The image description and the feature matching are performed by the extended version [1] of the content-base image retrieval system COMPASS [3] employed as object classifier. An image portion is described by means of histograms representing intensity, hue, saturation, edge distribution, while the Fourier coefficients and the Li Moments are used to describe its shape. The distributions of hue and saturation are represented also by two dimensional coocurrence matrices. The search process is very efficient, for example retrieving the closest items in a database of 1 million elements takes less than a second on a standard Pentium4 2GHz CPU.

The key point of the recognition algorithm is the computation of the linear cuts of a region Q in $\mathbf{R^2}$ (where Q is an object view in D or the region R). They are generated by occluding Q by $2n$ sheaves of parallel half-planes $\{\Sigma_i\}_{i=1,\ldots,m}$ with slopes in [0, 2π], where n and m are integer numbers called *generation cut parameters* (briefly GCPs). In particular, for each slope θ_k (k = 1, ..., n), the minimum bounding rectangle of Q with vertices P_0, P_1, P_2, P_3 is built (see Figure 1); a set $\{t_1,\ldots,t_m\}$ of real numbers in (0, 1) is fixed and for each i in $\{1,\ldots,m\}$ the rectangle with vertices $P_0+t_i(P_3-P_0)$, $P_1+t_i(P_3-P_1)$, P_2 and P_3 is computed. The Σ_i-linear cuts of Q are then $Q\cap P_0'P_1'P_2P_3$ and $Q-(Q\cap P_0'P_1'P_2P_3)$.

In this work, the slopes are equally spaced in [0, 2π], whereas the elements $t_1,\ldots,t_m$ are computed by taking in account the size of the database objects, as explained in Section 7.

The GCPs used in the object model construction can be different from those employed in the recognition algorithm and they are set by the user.

3 Object Model Construction

The user inputs are an object database D, the GCPs n and m for the linear cut generation, and two real values $p_a$and p_b in (0,1), $p_a < p_b$, related to the percentage of the minimum visible area of the linear cuts. We describe each object view O_v of D by low-level features, compute its area $A(O_v)$, occlude O_v by nm half-planes, and then insert O_v and its linear cuts having area in $[p_a A(O_v), p_b A(O_v)]$ in a new database D'. The linear cuts of O_v with area greater than $p_b A(O_v)$ are described and compared with the items of D, and they are put in D' only if the object classifier does not recognize them as instances of O, i.e. the most similar item of D' returned by the object classifier is not a view of O.
The parameters p_a and p_b are used to avoid the generation of too small linear cuts (not significant for recognition) and the insertion in D' of linear cuts very similar to the whole object view. Typically p_a ranges in (0, 0.6], while p_b is close to 1.

4 Heuristic Parameters

The algorithms for the recognition and reconstruction proposed in this work request a threshold Γ on the similarity measure and some heuristic geometric parameters (α_1, α_2, A_m, A_M for the recognition and $\langle E^A_{scale} \rangle$, $\langle E_{angle} \rangle$, a, b, y for the reconstruction).

The values α_1 and α_2 are user inputs and they define the range of variability of the scale factor for the database objects. Denoted by A_{min} and A_{max} the minimum and maximum areas of the elements of D', A_m is given by $\alpha_1^2 A_{min}$ and A_M is given by $\alpha_2^2 A_{max}$.

The values of Γ and $\langle E^A_{scale} \rangle$ are estimated as follows: for each view O_v of an object O of D, s rescaled and rotated versions of O_v are computed. The scale factors and the rotation angles are randomly chosen in $[\alpha_1, \alpha_2]$ and [0, 2π] respectively. For each transformed version T of (O_v), a linear cut is randomly generated and classified. If it is recognized as a cut K of a view of O, its scale factor α^* with respect to K is computed as square root between its area and the area of K. The error on scale is defined as the difference $|\alpha^* - \alpha_T|$, where α_T is the scale factor of the transformation T. The distributions of the recognition distances and of the errors on scale factor are computed. Γ is fixed as the 99th percentile of the distribution of the recognition distance, while $\langle E^A_{scale} \rangle$ is the mean value of the scale factors of the recognized objects. In this work, we consider 10 random transformations for each O_v (i.e. $s = 10$).

The value of $\langle E_{angle} \rangle$ measures the error on the angle in the reconstruction and it is defined as function of the GCP n (number of slopes) used for the object model construction, given by

$$\langle E_{angle} \rangle = \frac{1}{2} \frac{\pi}{n}. \tag{1}$$

The values a, b and y are related to the accuracy of the reconstruction and they are set by the user with the constraints $a, b, y > 0$ (see Sections 6 and 7).

5 Recognition Algorithm

The user inputs of this step are a region R of a color image, the GCPs (n_R, m_R) for the generation of the linear cuts of R, and the range $[\alpha_1, \alpha_2]$. The first step is the generation of the set $\{C_R\}$ of linear cuts of R. The cut C_R is a solution only if its area is in the range $[A_m, A_M]$, its scale factor α_R with respect to the most similar item K_R of D' belongs to $[\alpha_1, \alpha_2]$, and its recognition distance $d(C_R)$ is smaller than a threshold Γ. The scale factor α_R is computed by the formula

$$\alpha_R = \sqrt{\frac{A(C_R)}{A(K_R)}} \tag{2}$$

where $A(\cdot)$ indicates the area. The pairs (C_R, K_R) satisfying the conditions enumerated above are stored in the candidate list L and the most frequent object (if any) is returned as solution.
The complexity of the recognition algorithm is proportional to the feature extraction complexity (i.e. $O(A(C_R))$) multiplied the number of elements of the database D' and the number of linear cuts of R compared with the items of D'.

6 Reconstruction Algorithm

The reconstruction algorithm is based on a method of alignment between minimum bounding rectangles. Let (C_R, K_R) be a pair of the candidate list L with scale factor α_R. Let O_v be the object view whose K_R is a cut. The first step to reconstruct the whole shape of R is the determination of the function $\eta : K_R \mapsto C_R$ given by

$$\eta(x) = \frac{1}{\alpha}\mathscr{R}_\psi x + B(C_R) - B(\frac{1}{\alpha}\mathscr{R}_\psi K_R) \tag{3}$$

where x belongs to K_R, $\mathscr{R}$ is the matrix of the rotation of the angle ψ, and $B(\cdot)$ denotes the barycenter. The reconstruction of R is then given by $\overline{\eta}(O_v)$ where $\overline{\eta} : O_v \to \mathbf{R^2}$ is the extension of η on O_v.

The value of α is given by α_R. Let us determine the rotation angle ψ. Let a coordinate system be fixed. Let A, B, C, D and A', B', C', D' be the vertices (in clockwise or anticlockwise order) of the minimum bounding rectangles of C_R and $\alpha_R^{-1}K_R$ respectively (see Figure 2). Two cases are distinguished:

1. length(AB) $\neq$ length(AD) and length($A'B'$) $\neq$ length($A'D'$). Without loss of generality, we can suppose that length(AB) $<$ length(AD) and length($A'B'$) $<$ length($A'D'$). Since we do not consider the case of flipped objects, the possible transformations of the form (3) that align the rectangles $A'B'C'D'$ and $ABCD$ are two:

$$(A'B'', A'D') \mapsto (AB, AD) \text{ or } (A'B', A'D') \mapsto (DC, BC) \tag{4}$$

   Thus the angle ψ to be determined can assume the values $\theta_{A'B'} - \theta_{AB}$ or $2\pi + \theta_{A'B'} - \theta_{AB}$, where θ_{AB} and $\theta_{A'B'}$ are the directions of the edges AB and $A'B'$ respectively.

2. length(AB) = length(AD) and length($A'B'$) = length($A'D'$), i.e. the minimum bounding rectangles are squares. In this case there are four possible transformations and the angle ψ can assume the values $\theta_{A'B'} - \theta_{AB} + k\pi$ with k = 0, 1/2, 1, 3/2.

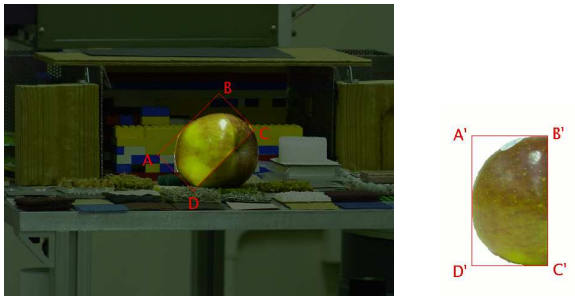


Figure 2: The cut of the apple in the image on left has been put on evidence by making darker the rest of the image. Its bounding rectangle ABCD has to be aligned to the rescaled bounding rectangle A'B'C'D' of the most similar element (on right).

Let $\Psi = \{\psi_i\}_{i\in I}$ be the set of the possible values of ψ.

Since the GCPs used in the recognition algorithm can differ from those employed in the object model construction, and the objects to be recognized can be rotated and/or rescaled, the values of α_R and ψ are affected by error. For this reason, we consider the two sets $S_\alpha = \{\alpha_j\}_{j\in\{0,\ldots,a\}}$ of possible scale factors and $S_\psi = \{\psi_{i,k}\}_{i\in I,k\in\{0,\ldots,b\}}$ of possible orientations, where

$$\alpha_j = \alpha_R \pm \frac{j}{a}\langle E_{scale}\rangle, \quad \psi_{i,k} = \psi_i \pm \frac{k}{b}\langle E_{angle}\rangle. \tag{5}$$

Then, we compute the functions $\eta_{j,i,k}$ aligning $A'B'C'D'$ and $ABCD$ and thus mapping K_R onto C_R, as in formula (3) with α_j in S_α and ψ_k in S_ψ.

For each pair (C_R, K_R), the functions $\eta_{j,i,k}$ are computed and extended to the functions $\overline{\eta}_{j,i,k}$: $O_v \to \mathbf{R^2}$. All the sets $\overline{\eta}_{j,i,k}(O_v)$ are the possible reconstructions of R. The best reconstruction of R is given by the function $\overline{\eta^*}_{j,i,k}(K_R)$ maximizing the *global overlap index* defined as

$$\frac{A(R\cap\overline{\eta}_{i,j,k}(K_R))}{A(R)} \tag{6}$$

and the *contour overlap index*, defined as the ratio $A(N_1\cap N_2)/A(N_1\cup N_2)$ where the regions N_1 and N_2 are defined as follows:

$$N_1 = \partial(\overline{\eta}_{j,i,k}(O_v))^w \cap \partial(\eta(K_R))^w, \quad N_2 = (\partial C_R \cap \partial R)^y. \tag{7}$$

The symbol ∂ denotes the topological boundary of the subsequent region, whereas the superscripts w and y indicate that the correspondent regions in the brackets are grown by w and y pixels respectively.

The parameter w is computed automatically and, as well as y, it is introduced because of the numerical approximations that occur in the computation of the function $\overline{\eta^*}_{i,j,k}$. In fact, the numerical approximations cause an imperfect overlap between the restriction of $\overline{\eta}_{i,j,k}(O_v)$ to $\eta_{i,j,k}(K_R)$ and $\eta_{i,j,k}(K_R)$. The difference between the two borders

$\partial(\overline{\eta}_{j,i,k}(O_v)) \cap \partial(\eta(K_R))$ and $\partial(C_R) \cap \partial(R)$ is generally very small, but it could be significant in the computation of the contour overlap index. For this reason, w is estimated by rounding up the tolerance

$$2\sqrt{\frac{A(\eta_{i,j,k}(K_R) \cap \overline{\eta}_{i,j,k}(O_v))}{A(\eta_{i,j,k}(K_R))}} \tag{8}$$

to the nearest integer.

On the contrary, the parameter y is set by the user and it is related to the accuracy on the reconstruction, that is measured by the overlap index $\nu = A(R \cap \overline{\eta^*}_{j,i,k}(O_v))/A(R)$. In our tests, we set $y = 2$ pixels.

The complexity of the reconstruction algorithm for a cut C_R is proportional to the product $4|I|abA(C_R)$, where $|I|$ is the cardinality of the set I.

7 Experiments and Conclusions

The section presents the experiments about the recognition and reconstruction performances obtained by running our method on 4 sets of synthetic images and on a set of real-world pictures. The tests on the synthetic data include some studies about the dependency of (*i*) the recognition rate on the GCPs used in model construction and in recognition algorithm, and (*ii*) the reconstruction accuracy on the parameters a and b.

Experiments on the Synthetic Data - Database IKEA-400 consists of 400 images of furniture (400 objects) of IKEA SpA downloaded from the internet catalog `http://www.ikea.com/ms/it_IT/our_products.html`.

The recognition and reconstruction performances have been tested on 4 datasets S_{15}, S_{25}, S_{35}, S_{45} containing rescaled and/or rotated versions of the objects of IKEA-400 partially occluded by synthetic patches (Figure 3). The subscript in the set names indicates the percentage of occluded area. The scale factor and the rotation angle range in [0.3; 1.5] and [0; 2π] respectively.

For the object model construction, we considered 10 sheaves of half-planes, i.e. 10 slopes equally spaced in [0, 2π]. Since the objects in the database have different size, the number of cuts having the same slope are different for each object and has been fixed by requiring that the minimum distance between two half-plane of the same sheaf is $\Delta = 50$ pixels. Moreover, $p_a = 0.40$ and $p_b = 0.90$. The resulting database IKEA-400' contains 10660 images. In particular, we obtained the following values: $\Gamma = 0.03574$, $\langle E_{scale} \rangle = 0.20044$, $\langle E_{angle} \rangle = 0.15708$, $A_m = 675.75$, and $A_M = 438898.5$ pixels. For the reconstruction we fixed $a = b = 2$.

We submit each visible part of the objects in the 4 test sets to the recognition algorithm with GCPs $n_R = 2$, $m_R = 2$ (case a.) and $n'_R = 5$, $m'_R = 5$ (case b.). Tables 1-a and 1-b show the recognition rate ρ and the reconstruction accuracy, measured by the mean overlap index $\overline{\nu}$ obtained by setting $a = b = 2$. In case a., the computational time for the recognition is less than 1 second, in case b. it is about 2.6 seconds, but the results are better. The recognition rate ρ is in case b. very satisfactory. For $a = b = 2$, the reconstruction requests about 6 seconds. For case b. we repeated the tests by fixing $a = b = 4$ and we obtained the mean overlap index $\overline{\nu}'$. There is not a significant difference between the values of $\overline{\nu}$ and $\overline{\nu}'$, but for $a = b = 4$, the computational time is longer (about 20 seconds). The times have been measured by running the algorithm on a standard Pentium4 2.80GHz.

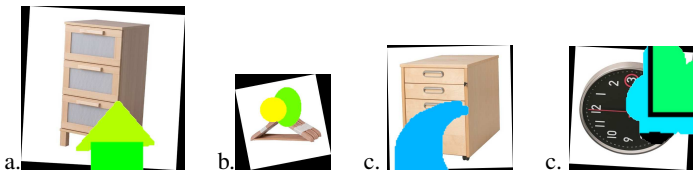


Figure 3: Examples of synthetic test images. Each of them contains a rescaled and/or rotated object view of D partially occluded (occluded area percentages of a. 15%, b. 25%, c. 35 %, and d. 45%).

Table a

Test Set	ρ	$\overline{\nu}$
S_{15}	0.865	0.756
S_{25}	0.748	0.725
S_{35}	0.798	0.666
S_{45}	0.668	0.626

Table b

Test Set	ρ	$\overline{\nu}$	$\overline{\nu}'$
S_{15}	0.983	0.882	0.892
S_{25}	0.885	0.848	0.852
S_{35}	0.965	0.815	0.821
S_{45}	0.955	0.796	0.821

Table c

Test Set	ρ	$\overline{\nu}$
S_{15}	0.980	0.876
S_{25}	0.876	0.814
S_{35}	0.945	0.799
S_{45}	0.918	0.762

Table 1: Recognition rate and mean overlap index for the test sets with objects of IKEA-400. The overlap indices ν and ν' have been obtained by setting ($a = 2$, $b = 2$) and ($a = 4$, $b = 4$) respectively. Tables a. and b. are referred to the database IKEA-400', whereas the results of Table c. have been obtained by using the database IKEA-400".

To test the results dependency on the GCPs used for the object model construction, we built an other extension of IKEA-400 by generating cuts with 5 sheaves of half-planes with $\Delta = 100$ pixels. The resulting database (IKEA-400") contains 5573 images. Table 1-c shows the recognition rate and the mean overlap index obtained by using in the recognition algorithm the GCPs $n'_R = m'_R = 5$ and the parameters $a = b = 2$ for the reconstruction. Also in this case satisfactory results have been obtained.

Experiments on the Real-World Data - The tests on the recognition and reconstruction performances of our method on real-world images have been carried out by considering a database of 8 objects (that we denote with 3DOBJ) along with a ground-truth of 51 pictures containing whole or partially occluded objects of 3DOBJ [15]. This material is freely available on `http://tev.itc.it/DATABASES/objectsPonce.html`. The database 3DOBJ consists of 161 images of 8 objects, each represented by some 2D views. The database 3DOBJ' extended by the cuts has been built by considering 10 sheaves of half-planes, each of them - as well as in the case of the synthetic data - contains a number of half-planes different for each objects. The values of p_a and p_b are 0.20 and 0.90 respectively. The heuristic thresholds have the following values: $\Gamma = 0.02493$, $\langle E_{scale} \rangle = 0.20104$, $\langle E_{angle} \rangle = 0.15708$, $A_m = 442.35$, and $A_M = 250762.5$ pixels. The parameters for the reconstruction are $a = 5$, $b = 7$. We selected by a semi-automatic segmentation the visible parts of the objects in the test images and we classify them by setting $n_R = m_R = 10$. The obtained recognition rate is 80.89%. The most of cases of non recognition are due to the fact that the objects in the test images are differently illuminated with respect to the object views in the database and our features are not invariant by changing the light conditions. The reconstruction is in this case harder, because in the test images, the objects to be recognized appear under different points of view respect to the shots stored in the database. This is the case of the shoe shown in Figure 4. Nevertheless, the mean

Figure 4: The set of views of the shoe in the database D (left) does not contain the pose of the test image on right. In this case, the object is recognized, but a good reconstruction is not possible.

Figure 5: Examples of Reconstruction in Real-World Images.

overlap index is 0.9280. Some results are shown in Figure 5.

Conclusions - The tests illustrated in this Section show that our method performs good both in term of recognition and reconstruction. Therefore our future work includes its use in an object recognition system like [10]. Since the computational times are quite long, our planes comprehend also a parallelization and optimization of the code and the comparison of our approach with other on common databases. Moreover, we will develop a strategy for the automatic estimation of the thresholds that are currently to be fixed by the user, like the GCPs parameters.

Acknowledgments - The authors would like to acknowledge IKEA SpA for having made available its catalogs for scientific research purposes. This work has been partially supported by the EU Project VIKEF (*Virtual Information and Knowledge Environment Framework*).

References

[1] C. Andreatta. CBIR techniques for object recognition. Technical Report, ITC-irst T04-12-01, December 2004.

[2] S. Berretti, A. Del Bimbo, and E. Vicario. Efficient matching and indexing of graph models in content-based retrieval. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 23(10), 2001.

[3] R. Brunelli and O. Mich. Image retrieval by examples. *IEEE Transactions on Multimedia*, 2(3), 2000.

[4] T. Deselaers, D. Keysers, R. Paredes, E. Vidal, and H. Ney. Local representation for multi-object recognition. In *Deutsche Arbeitsgemeinschaft für Mustererkennung: DAGM 2003*, 2003.

[5] H. Eidenberger. A new perspective on visual information retrieval. In *SPIE Electronic Imaging Symposium, SPIE vol. 5304*, San Jose, 2004.

[6] V. Ferrari, T. Tuytelaars, and L. Van Gool. Simultaneous object recognition and segmentation by image exploration. *International Journal of Computer Vision (IJCV)*, (4), 2006.

[7] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, H. Qian, and other. Query by image and video content: the QBIC system. *Computer*, 28(9), 1995.

[8] G. Fritz, L. Paletta, and H. Bischof. Object representation and recognition from informative local appearances. In *Digital Imaging in Media and Education*, 2004.

[9] F. Krolupper. Recognition of occluded objects using curvature. In *Proc. of The 12th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2004.

[10] M. Lecca. Object recognition in color images by the self configuring system MEMORI. *International Journal of Signal Processing*, 3(3), 2006.

[11] Philippe Mulhem and Joo Hwee Lim. Symbolic photograph content-based retrieval. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, New York, NY, USA, 2002. ACM Press.

[12] S. Obdrzlek and J. Matas. Object recognition using local affine frames on distinguished regions. In *British Machine Vision Conference*, 2002.

[13] M. Reinhold, M. Grzegorzek, and H. Niemann J. Denzler. Appearance-based recognition of 3-d objects by cluttered background and occlusions. *Pattern Recognition*, 38(5), 2005.

[14] E. Rivlin, S. J. Dickinson, and A. Rosenfeld. Recognition by functional parts. *Computer Vision and Image Understanding: CVIU*, 82(2), 1995.

[15] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. 3D object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *IJCV*, 66(3), 2006.

[16] A. Shokoufandeh, I. Marsic, and S. J. Dickinson. View-based object recognition using saliency maps. *Image and Computing*, 17, 1999.

[17] J. R. Smith and S.F. Chang. VisualSEEk: A fully automated content-based image query system. In *ACM Multimedia*, 1996.

[18] V. Vilaplana, X. Giro, P. Salembier, and F. Marques. Region-based extraction and analysis of visual object information. In *Proc. of Int. Workshop on Content-Based Multimedia Indexing CBMI 2005*, 2005.

Colour Constancy Based on Model Selection

Chuan Li, Peter M. Hall
Department of Computer Science
University of Bath

Abstract

Colour constancy is the problem of mapping colours in a scene as the lighting environment changes. This problem has been studied for many years, and linear solutions that give excellent results are available for Lambertian scenes. More complex reflective environments demand non-linear transforms, but the models are harder to fit and in some cases bring worse results due to over-fitting. We introduce a fully non-linear colour constancy model and a method to select an optimal model from amongst a family of discrete models, including linear models. Once given a set of colour matches our model selection is fully automatic. The selection method is experimentally verified, and is put to use in two applications: scene re-lighting and environment mapping.

1 Introduction

The goal of colour constancy is to achieve consistent colour relationships as illumination changes. This requires mapping a source colour distribution to a target. Early work from Forsyth [9, 10] independently scaled each component of a source colour $\mathbf{x} = (x_1, x_2, x_3)$ to obtain a target $y_i = \alpha_i x_i$; this is the so-called "diagonal" model (also called von Kries model) because the scale coefficients occupy the diagonal of a 3×3 matrix. These α are selected by model fitting. Finlayson showed that a 2-dimensional colour gamut (division by blue) makes for easier fitting [5] [4] [6] [8]. Later, he and others also showed the accuracy for diagonal model can be improved [7] [11], especially via a sharpening transformation [12].

The diagonal model is of benefit when an automated solution is required. However, the accuracy of diagonal transforms can be improved upon using general linear transform model [15]. Even more using non-linear transforms [14], these authors restrict themselves to quadratic transforms of the forms $y_i = \gamma_i + \sum_{j=1}^{3}(\alpha_{ij}x_j^2 + \beta_{ij}x_j)$. We refer to this as the "diagonal quadratic" model because they do not have cross items x_1x_2, x_2x_3 and x_3x_1. The cost of improved accuracy is the need for a greater number of colour matches between source and target, often available only via user interaction. This strongly suggests that task plays an important role in the selection of an appropriate colour transform. Yet task is not the only governing criteria: the optimally accurate transform depends on scene content as much as on illumination change. Otherwise said, there is no single transform that is accurate for all objects in a given scene [3].

To see this consider the following example. We cannot distinguish red from white when viewed under a red light. Of course, both will be transformed to the same target

colour. This is against the fact, as we can differ these two colour in any other lights that are not pure red. The aim of colour constancy is therefore to find an optimal model for particular scenes.

Our work makes the following two contributions:

- we introduce a complete quadric transform including crossed items of the form $x_i x_j$ to give a general quadratic model.
- we select the optimal model from all those available (diagonal, general linear, diagonal quadratic and general quadratic) rather than adhere to any single one.

The general model we use can be written as a tensor form: $y_i = [\mathbf{x};1]^T \mathbf{Q}_i [\mathbf{x};1]$; $i = 1:3$ are for red, green and blue channel respectively. We also bring in the homogeneous coordinate (appending 1) in order to have the linear part of the transform. Each layer of the tensor $\mathbf{Q}$ is a 4×4 triangular matrix, denoted $\mathbf{Q}_i$, that represents the mapping for a colour channel from source to target. Model fitting requires estimating all of the elements in $\mathbf{Q}$, given a set of sample matches.

In this paper we consider four models: diagonal linear, general linear, diagonal quadratic and general quadratic (DL, GL, DQ, GQ for short). For all these models, we constrained the entry $\mathbf{Q}_{i44}$ to be zero. This means we don't translate all colours by the same amount (Experiments show translation plays an insignificant role in colour constancy problem). Given a coloured image and a set of colour matches, our model selection strategy selects the optimal model (see Section 2). The strategy takes the quality of the matches into account: quality being related to the distribution of the coloured matches within the distribution of the image as a whole. The rationale behind our selection mechanism is given empirical support by experiments described in Section 3. The research in this paper was motivated by the two applications we describe in Section 4. The first application is colour modification of scene for which no target exists — as if the scene were re-lit under a different illuminant. We show how matches from an example source and target pair taken from a different scene can be used to re-light the given, novel scene — this includes picking the best model. The second application is environment mapping in which we create a synthetic reflection in a coloured reflector — a coffee pot in our case.

2 Model Selection

Our model selection picks the best fitting model for a specific scene based on n colour matches $(\mathbf{x}_j, \mathbf{y}_j)$, provided by user interaction. The key to our approach is to consider both the fitting error λ and stability ν for each model. We find the model that minimises some linear combination of these two.

Fitting error, λ, is the distance between the transformed colours of source sample and their corresponding target colours in RGB space. In this case the transform is based on the full set of samples given by the user. More specifically, we define λ for each channel $i = [1,3]$.

$$\lambda_i = \frac{1}{n}\sum_{j=1}^{n}(\mathbf{x}_j^T \mathbf{Q}_i \mathbf{x}_j - y_{ij}) \tag{1}$$

This allow us to select different model for different colour channels.

Stability, ν, on the other hand, is computed from a collection of images $g^k(\mathbf{x})$. Each image is the transform of the full source image, but now based on the kth subset of the given matches; this is a "leave one out" strategy.

$$\nu_i = \frac{1}{|S|}\sum_{x\in S}\left(\frac{1}{N}\sum_{k=1}^{N}(g_i^k(\mathbf{x}) - \bar{g}(\mathbf{x}))\right) \tag{2}$$

in which N is the number of subsets chosen and S is the spatial domain of the source image, $\bar{g}(\mathbf{x})$ is the average of all transformed images.

A standard way to select a model given accuracy and variance is to defined an "energy" as their sum: $\xi = \lambda + \nu$ (see [2]). We found this to be insufficient, because of the quality of the samples. It is well known that a model fitted to tightly spaced points often give a poor account for distant point. This implies we will observe a high variance, if we successively select subsets of tightly points. This effect is amplified for non-linear models when compared to linear models. Our model selection strategy takes this into account: it prefers linear models when the given samples a tightly spaced, when compared to the distribution of colour in the image as a whole.

The optimal model for the ith colour channel minimises ξ given by

$$\xi_i = w\lambda_i + (1-w)\nu_i \tag{3}$$

Here w is a weight. The role of w is to give some allowance for the "quality" of the sample points. Larger w means higher quality. This linear combination indicates when the quality of the sample is low, ξ is mainly decided by stability ν; when the quality of the sample is high, we credit more on the fitting error λ. This follows our intuition and has been experimentally proved in Section 3. (see Figure 2)

We designed two forms of weights for different applications. An easy way to choose w is via the ratio r of convex hulls: the volume contained by the samples to the volume of all colours in a scene. This definition works well when all the colours in samples belong to colours in the scene. However in the applications such as scene re-lighting, we aim to use colour samples from one scene to reproduce the illumination change for another. In general the colour distributions for the source image of scene A and the image of scene B will differ, so we are likely to have colours sampled from A that don't exist in B. In this case, we redefined r as $r = L/\rho$, in which ρ is the mean of the distance between all the colours in B and their nearest neighbours in the samples taken from A. The constant L is set in advance, we have found $L = 0.01$ is a suitable value (see Figure 4 and here we assume colour components in the range $[0,1]$).

However, we found that simply setting $w = r$ gave poor results — model switching is better served when w is a sigmoid function of r, specifically:

$$w = \frac{1}{1+\exp(u(c-r))} \tag{4}$$

The values for the cutoff $c = 0.7$ and the gain $u = 20$ were decided by experiment (Figure 2 and 4).

A note on estimating the parameters of a given model. This is easily done by factoring out the $\mathbf{Q}$ as a vector to obtain the homogeneous linear system $\mathbf{XQ} = 0$. The design matrix $\mathbf{X}$ comprises rows built from matching pairs. The fully quadratic model uses rows in the

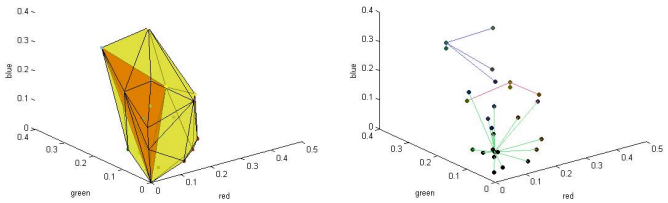


Figure 1: Two ways to estimate the sample quality. Left, the sample quality is a function of the ratio r of convex hulls: the volume contained by the samples to the volume of all colours in a scene. Right, the sample quality is a function of the mean of the distance between all the colours in the scene and their nearest neighbours in the samples. Here the sample has 3 different colour matches.

design matrix of the form

$$[x_1x_1,\ x_2x_2,\ x_3x_3,\ x_1x_2,\ x_1x_3,\ x_2x_3,\ x_1,\ x_2,\ x_3,\ -y]$$

for each colour channel and for each matching pair. Less general models use design matrices with appropriate terms removed. The aim is to ensure a design matrix with the smallest possible number of columns, and with at least as many rows as columns. The singular value decomposition of the design matrix then yields a solution in the least squares sense.

3 Experiment

This section details our experiment to show that the optimal model for a given scene depends on the quality of the samples.

Recall we constrained the candidate model into four forms: diagonal linear, general linear model, diagonal quadratic and general quadratic. These models have different degrees of freedom but as long as we have sufficient numbers of colour matches, we can find least square solutions for them. The image we use are from colour constancy test database collected by the colour group at Simon Fraser University [1]. In this database the scenes have been set up such that only the colour of illumination changes. Hence images from the same scene can be used as the ground truth to each other.

Given same colour matches, we estimate the transform $\mathbf{Q}^k$ for each of the four candidate models M^k — we make no choice of model during the experiment. Note that superscripts index models so $k \in [1,4]$; subscripts $\mathbf{Q}_i$ index colour channels. Having estimated a transform we apply it to transform the colours in a source image $f(\mathbf{x})$ to obtain image $g^k(\mathbf{x})$. The rms error between $g^k(\mathbf{x})$ and the ground truth target picture $h(\mathbf{x})$ indicates the fitting accuracy.

To make a randomised trial the experimenter chooses a large number m of samples as the full sample set. In our experiment this full sample set contains 24 colour matches from the *GretagMacbeth ColorChecker*^{TM} [1] chart. Then we randomly select numbers of matches from them to estimate transformations for each model. Because the

[1] `http://www.cs.sfu.ca/čolour/data`

general quadratic model has the highest degree of freedom 9, so each selected sample subset should at least contains 9 matches. Our test images cover 10 different illumination changes of the same scene. For each illumination change, we select 1600 subsets containing matches of number from 9 to 24. Using each sample subset, we compute $\mathbf{Q}^k$ for M^k. Figure 2 shows the relationship between the sample's quality and the models' performance. We use both definitions of weight to analysis the result.

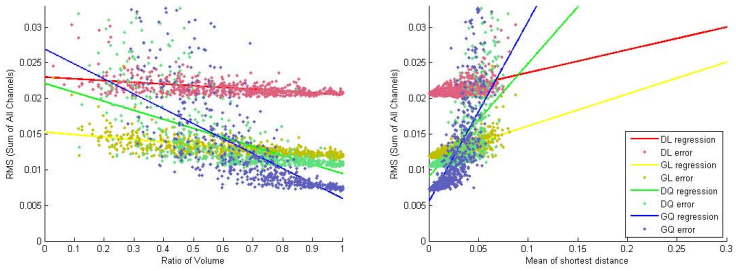


Figure 2: Error grows as sample quality falls. Left: the ratio of convex hulls measures sample quality, larger ratios means better quality. Right: using the mean of the distance between all the colours in the scene and their nearest neighbours in the samples, shorter is better.

From Figure 2 we can see there is a overall decrease of fitting error for all models when the ratio of convex hulls rises or the mean nearest neighbour distance falls. And compared to simple models such as linear ones, the advantage of complex models become much more significant when the quality is high enough (weight approaches 1). On the other hand, the complex models tend to be very unstable when weight is close to zero. This means quality of samples influences the performance of models, especially those nonlinear ones.

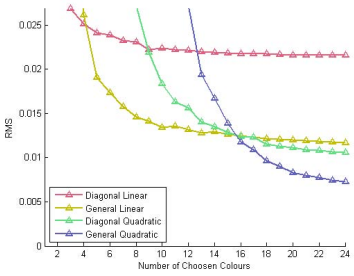


Figure 3: Error as a function of the number of samples. When the number of samples rises, more complex models can be chosen.

Based on this, we design the combination of λ and ν based as an interpolation by weight w. Though this combination is as simple as linear, the effect of w has been pre-

adjusted by the sigmoid function (see Equation 4). And through $cutoff$ and *gain*, we can control the manner of model selection, for example, large cutoff is conservative and small cutoff is greedy.

Though in Figure 2 the GL model (yellow) seems always better then the DL model, this is because we at least have 9 colour matches to estimate the transform. This situation will change if the number of matches falls: the degrees of freedom of a model gives a lower bound on the number of matches. As shown in Figure 3, the DL model (red line) is best when the number of matches is less than 4.

Figure 4 shows the result of our model selection. With increasing sample quality, our model selection result switches from DL to GQ at a volume ratio of about 0.7, which is the cutoff in Equation 4. This value is suggested by Figure 3 and been confirmed by extensive testing across many scenes.

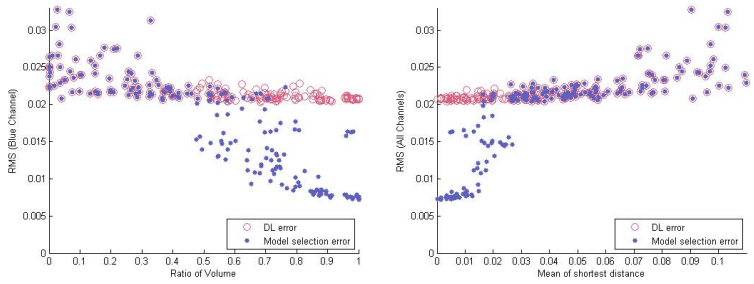


Figure 4: The results of model selection. Left, the result given by weighting the sample quality via ratio of convex hulls. Right, the result given by weighting the sample quality by the mean of the distance between all the colours in the scene and their nearest neighbours in the samples.

Note that although the results presented above are typical, they come from a single scene. We have already commented that model selection is scene depended. Therefore, we cannot conclude that any model is redundant. Our philosophy is that the DL model is the most stable but can be improved using other models if conditions support a switch. Hence conservative choice of parameters in Equation 4 keeps us save.

4 Applications

We used our colour model selection method in two applications. The first one is scene re-lighting and the second one is environment mapping into coloured reflectors.

4.1 Scene Re-lighting

Scene re-lighting as we intend involves transforming the colours in an image to give the appearance of an illuminance change in the real world scene. We limited changes of illuminant to colour only —we do not consider changes in a light's spatial location, for example. To re-light a scene B we need to know the colour changes for a subset of its

colours, but we will have no matches — the target picture is to be synthesised. So we use set of matches taken from from other scene, scene A, for which both a source and target set exist.

The simplest approach is to use the matching samples between the source and target for scene A to compute a transform for scene B. However, we know that an optimal transform is scene dependent. Our application attempts to do chose an optimal transform for the scene B, given samples taken from the source and target of scene A. This is possible because accuracy λ is computed from the samples alone, but variance ν is computed from the set of transformed images. Since we can always transform scene B, we can compute a variance, and the existence of the sample means we can compute accuracy.

As previously mentioned (in Section 2), in general the colour distributions for the source image of scene A and the image of scene B will differ. So here we use the second definition of the weight: a function of the mean of the distance between all the colours in B and their nearest neighbours in the samples taken from A. Figure 5 shows the best form of transform can vary from scene to scene, and our model selection can help us to choose the best one.

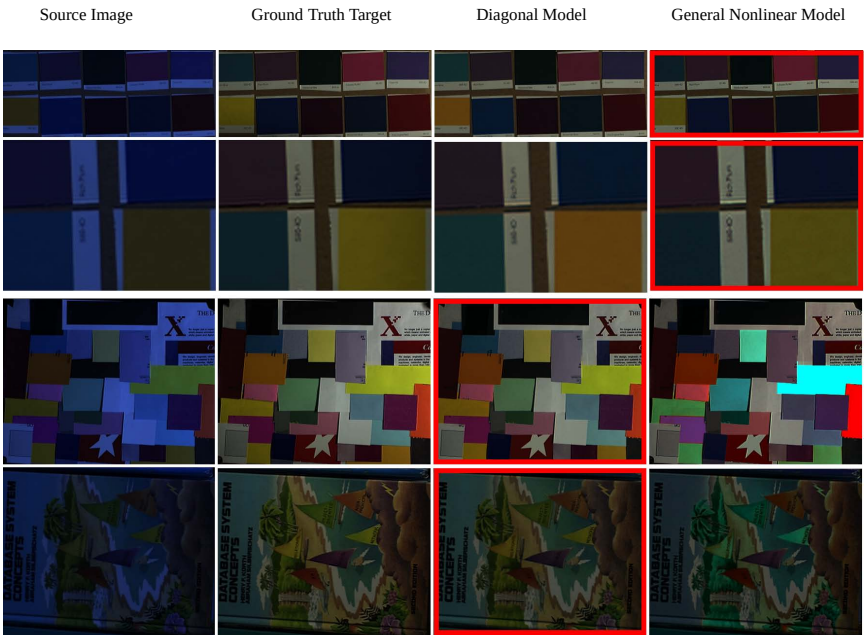


Figure 5: Result of different models for scene re-lighting. Model selection results are indicated by red rectangles. We sampled colours from the first scene (the top row) and use them to produce the same illumination change for different scenes. When the scene can be well described by samples (the first row an second rows), the complex model gives better results. When sample quality is low according to the scene (the third and fourth rows), the simple model tends to be stable. In both cases, our model selection strategy chose the best models.

4.2 Environment Mapping

Inserting graphical objects into real world imagery is a contemporary aim that arises in many applications. There are many interesting problems, here we focus on reflecting synthetic scenes in mirrors, which is one kind of environment mapping. The mirror in our example happens to be both coloured and curved, which make the problem even more interesting. A full discussion of reflecting into curved visual objects (without reconstruction in three dimensions) would take us beyond the bounds of this paper, but see [13]. Here we focus on the colour transform problem.

Figure 6: Non-linearity in specular reflection. Bottom row shows the re-lit coffee pot. The left-most pair were created using low quality samples, The right-most pair using high-quality samples. The left of each pair is the result of a diagonal linear model, the right a general quadratic model. The diagonal model is selected for low quality samples, the general quadratic for high quality samples.

The need for general non-linearity is suggested in Figure 6. It shows a silver coffee pot and a green Christmas bauble, captured under identical conditions. The reflections of these two reflectors are both specular. We aim to accurately transform the coffee pot's silver surface to the bauble's green surface (including the reflection). To achieve this, sufficient colour samples and a non-linear transform are needed. This is because the highlights remain approximately a constant colour regardless of the colour of the reflector, but coloured regions away from highlights change to a considerable degree. Figure 6 shows the results given by transforms estimated from different samples and different models. The best result is from general quadratic transform when samples are sufficient, which has been successfully picked up by our model selection strategy. However, when samples are not sufficient, our model selection can also choose the less complex transforms.

Environment mapping is not confined to changing the colour of reflectors, as in Figure 6. It is possible to insert purely synthetic imagery into the reflector and change the

reflector's colour. In this case, we also need to consider the stability of the transform for the insert. This can be easily achieved via considering colour of the insert during computing variance ν. Figure 7 shows how we insert new object into reflectors, and re-lighting the synthetic reflections. In this case a diagonal model was selected. This is because the insertion changes the colour gamut from the colour calibration pictures in Figure 6. Hence, in this case, samples are of lower quality and a stable model is selected

Figure 7: Environment mapping into a coloured reflector. The first picture is the calibration scene. The second picture is a insert object. The third picture is the mapping result for the insert – its colour was transformed to match the silver surface of the coffee pot and its shape was non-linear warped. The fourth picture is the synthetic reflection. The fifth picture is the result of re-lighting the synthetic reflection to the green bauble in Figure 6 .

5 Conclusion

We have introduced a fully non-linear colour constancy model and a method to automatically select an optimal model for a given scene. The non-linear model and selection method have been used in two applications. Each application forced novelty of its own: in scene re-lighting we changed the colours of a second picture based on samples taken from an example pair thereby forcing a redefinition of sample quality; in environment mapping we had to account for the non-linearity brought about by specular reflections which motivated the general non-linear model.

Our work has lead us to the following conclusions. First, that although colour constancy has no generally optimal solution, specific solutions can be found. Second, that the optimal model for a specific solution can be selected automatically. The particular model chosen critically depends on the quality of the samples used to calibrate the mapping. Third, non-linear mappings are of value in practical applications such as environment mapping.

We do rely on interaction in that as users must input initial matches. This gives us the space for future work – to produce an fully automatic system. This is challenging because scene content can change dramatically, hence colour matches between them are difficult to be found. But as to applications such as environment mapping, the users will benefit from a fully automatic system as it can reduce their workload.

6 Acknowledgements

This work was supported by EPSRC grant. The authors also gratefully acknowledge the colour group at Simon Fraser University for opening their collection of test database.

References

[1] *GretagMacbeth Color Management Solutions*. http:// www.gretagmacbeth.com.

[2] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley Interscience, 2001.

[3] M. D. Fairchild. *Color Appearance Models*. Addison Wesley, 1997.

[4] G. D. Finalyson. Color in perspective. In *IEEE Trans. Pattern Analysis and Machine Intelligence*, volume 18, pages 1034–1038, Washington, DC, USA, 1996. IEEE Computer Society.

[5] G. Finlayson. *Coefficient color constancy*. PhD thesis, Simon Fraser Univ., School of Computing, Burnaby, BC, Canada., 1995.

[6] G. Finlayson and S. Hordley. A theory of selection for gamut mapping color constancy. In *CVPR*, volume 00, page 60, Los Alamitos, CA, USA, 1998. IEEE Computer Society.

[7] G.D. Finlayson, M.S. Drew, and B.V. Funt. Diagonal transforms suffice for color constancy. In *ICCV*, pages 164–171, 1993.

[8] G.D. Finlayson, B.V. Funt, and K. Barnard. Color constancy under varying illumination. In *ICCV*, volume 00, page 720, 1995.

[9] D. A. Forsyth. Colour constancy and its applications in machine vision. *PhD thesis, Oxford University, Department of Engineering Science*, 1988.

[10] D. A. Forsyth. A novel algorithm for color constancy. In *IJCV*, volume 5, pages 5–36, Hingham, MA, USA, 1990. Kluwer Academic Publishers.

[11] B. V. Funt and Hao Jiang. Nondiagonal color correction. In *ICIP*, pages 481–484, 2003.

[12] M. Drew G.D. Finlayson and B. Funt. Spectral sharpening: Sensor transformations for improved color constancy. In *Journal of Optical Society of America*, volume 11, pages 1553–1563, 1994.

[13] P.M. Hall, J.P. Collomosse, Y.Z. Song, P.Y. Shen, and C. Li. Rtcams: a new perspective on non-photorealistic rendering from photographs. to appear in IEEE Transactions on Visualization & Computer Graphics.

[14] A. Ilie and G. Welch. Ensuring color consistency across multiple cameras. In *ICCV*, pages 1268–1275, Washington, DC, USA, 2005. IEEE Computer Society.

[15] N. Joshi. Color calibration for arrays of inexpensive image sensors. Master's thesis, Stanford University Department of Computer Science, 2004.

Interest-point Based Face Recognition from Range Images

F. R. Al-Osaimi, M. Bennamoun and A. Mian
The University of Western Australia
35 Stirling Highway, Crawley, WA 6009, Australia

Abstract

We present a novel approach to interest-point detection tailored to range images. A range image is represented by two images with blob-like patterns that have easily detectable peaks and can be efficiently extracted using convolution kernels. These kernels were designed to produce repeatable and independent blob-like patterns when convolved with the range image. The interest-points correspond to peaks of the patterns after dropping the unstable ones and performing Non-Maximal Suppression (NMS) on their union. The approach was applied to facial range images from the FRGC V2.0 dataset and about 88% repeatability was achieved. Face recognition was also performed by matching the local range regions around the interest-points. An approach based on three levels of matching combined with RANSAC algorithm was used to increase the correct matches and reduce the false ones. Preliminary recognition results for a database of 466 subjects and 1765 probes were 96.33% identification rate and 90% verification rate at 0.1% False Accept Rate (FAR) for faces under neutral expression.

1 Introduction

In recent years, the paradigm of object recognition by matching local regions around interest-points (point features) has been the focus of research in computer vision, especially in 2D recognition. This paradigm has many vital advantages over the classical recognition approaches. For example, it is robust to occlusions and does not require object/background segmentation. We believe that it can be also advantageous in the context of 3D face recognition as it has been shown that recognition by matching local regions [1] or point features [5] is more robust to makeup and facial expressions. However, applying this paradigm to 3D face recognition requires a suitable interest-point detection approach.

In spite of that, some approaches to 3D face recognition are based on matching local regions. In the approach by Moreno et al. [3] local regions are segmented according to the mean and the Gaussian curvatures and the segmented regions are then matched against each other. Errors in the curvatures (which are sensitive to noise) may affect the segmentation of the local regions. Elastic Bunch Graph Matching (EBGM) which matches local regions was extended to face recognition from integrated 2D and 3D images [4]. Although EBGM is a successful face matcher, it is based on a fixed small number of points. Consequently, it is not as robust as interest-point based recognition. In the approach by Mpiperis et al. [5] local features are computed around all the 3D points.

It is tempting to apply the existing interest-point detection approaches that are designed for 2D images to range images (e.g. the SIFT [6]). However, the nature of 2D images is different from range images and this difference can affect the applicability of these approaches to range images. Firstly, these existing approaches usually rely on salient features which are usually induced by texture such as edges, corners and/or blobs. On the other hand, pixels (range values) of range images smoothly vary. Consequently, range images may not have sufficient easily detectable interest-points. For example, there is a very limited number of such features in a range image of a human face. Applying a 2D image interest-point detection technique to such range image may result in either an insufficient number of interest-points or unreliable ones depending on the selection of the tuning parameters. Secondly, under rigid transformations, the appearance of an object in range images vary differently from its appearance in 2D images. In a 2D image the value of a pixel representing a point on the object generally remains constant with rigid transformations (apart from the effects of illumination) whilst the pixel value in range images varies accordingly.

In our approach to interest-point detection for range images, we represent the range image by multiple images of blob-like patterns from which a sufficient number of repeatable interest-points can easily be detected at the peaks of these patterns. The patterns in the representing images are independent from each other in the sense that they can define different interest-points based on different 3D surface information. The representing images are efficiently extracted by convolving the range image with kernels of sufficiently large sizes and non-overlapping spatial spectrums. Such kernel sizes suit range images especially when the range image lacks sufficient salient features as they cover larger pixel neighborhoods and generate response based on more surface information. In addition, the kernels are robust to object translations and rotations within $\pm 15^{\circ}$.

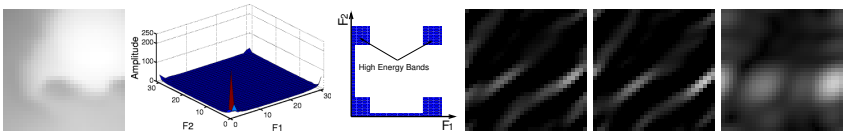
2 Input Range Images

The input range images are computed from the frontal facial 3D pointclouds of the FRGC v.02 dataset [7]. The data is in the form of three matrices x, y and z. The spikes are removed by dropping the outlier points from the three matrices based on local statistics. The matrices are then smoothed using a mean filter which neglects the missing points. After that the holes are filled using bi-cubic interpolation of the missing points. The range image was computed from the three matrices by interpolating for integral x and y coordinates and storing the corresponding z coordinates in the range image matrix using x as horizontal index and y as a vertical index. Finally, the range image is smoothed using a Gaussian filter.

3 Interest-points Detection for Range Images

The summation of a group of adjacent spatial frequencies forms spatial beats in a similar manner to the well-known sound beats phenomenon, as the differences in the spatial frequencies also cause repeated patterns of constructive and destructive interferences over spatial distances. Depending on the phase and amplitude values of the spatial frequencies, the spatial beats can shift, intensify and/or merge. From these spatial beats, the representing 2D blob-like images can be computed (see Fig. 1).

According to 2D Discrete Fourier Transform (DFT), a range image $I(x,y)$ of size $N \times M$ can be represented as the summation of complex exponents as in the well-known



(a) (b) (c) (d) (e) (f)

Figure 1: (a) is a window from a facial range image (part of the nose appears in the widow). (b) Discrete Fourier transform (DFT) of the range window. (c) shows the frequency bands which have high energy. (d) and (e) are real and imaginary spatial beats generated by taking inverse DFT of a subset of 3×3 adjacent frequencies. (f) a window of blob-like patterns that was generated by taking the absolute values of (d) and (e).

Eqn. 1.

$$I(x,y) = \sum_{m=0}^{M-1}\sum_{N=0}^{N-1} a_{mn} e^{j2\pi(mf_1x+nf_2y+\phi_{mn})}$$
$$= \sum_{m=0}^{M-1}\sum_{N=0}^{N-1} a_{mn}\cos(2\pi(mf_1x+nf_2y+\phi_{mn})) + j\sum_{m=0}^{M-1}\sum_{N=0}^{N-1} a_{mn}\sin(2\pi(mf_1x+nf_2y+\phi_{mn})) \quad (1)$$

where $f_1 = \frac{1}{M}$ and $f_2 = \frac{1}{N}$ are the fundamental horizontal and vertical frequencies, respectively. The factors a_{mn} and ϕ_{mn} are the amplitudes and the phase shifts of the spatial frequencies.

By taking square windows of different sizes (from 21×21mm to 31×31mm) from many facial range images and examining their spatial spectrum using DFT, only certain frequency bands have high energy (See Fig. 1.c). These frequency bands are the frequencies which are low in both directions (LB), the ones which are horizontally low and vertically high (LH), the ones which are horizontally high and vertically low (HL), the ones which are high both vertically and horizontally (HB), the stripe of frequencies that are horizontally low (HS), the stripe of frequencies that are vertically high (VS). We are interested in generating a blob-like image from a band of adjacent frequencies (a window 3×3 frequencies) that have sufficient energies and are less affected by rigid transformations. The HB band and the frequency subset of the LB band which are highest horizontally and vertically seem more appealing as it can be shown empirically that the frequencies which are lowest either horizontally or vertically are generally more affected by the orientation of a given surface (pitch and yaw rotations).

3.1 Kernel Design

The selected 3×3 window of the adjacent spatial frequencies $\mathscr{F}$ produces a corresponding signal R that has real and imaginary spatial beats according to Eqn. 2.

$$R(x,y) = \sum_{(a_i,\phi_i)\in\mathscr{F}} a_i\cos(2\pi(m_if_1x+n_if_2y+\phi_i)) + j\sum_{(a_i,\phi_i)\in\mathscr{F}} a_i\sin(2\pi(m_if_1x+n_if_2y+\phi_i)) \quad (2)$$

The blob-like image can be extracted from R by simply taking the absolute value of R (see Fig. 1.f). An equivalent but more efficient way to achieve that is to convolve the range image with a corresponding kernel r. The kernel r was computed by setting the selected frequencies set $\mathscr{F}$ to 1 and all the other frequencies to 0 then taking the inverse DFT. In a similar way, the blob-like image can be computed by taking the absolute value of the convolution of the range image and r.

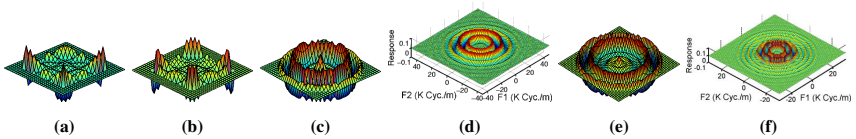


Figure 2: (a) and (b) are the real and imaginary components of the kernel that passes the selected adjacent frequencies. Their corners have high weights which affect their invariance to rotation around z axis. (c) and (e) are rotationally invariant kernels computed from the real kernel (a) by rotating it in small steps from angle 0° to 360°. The interpolation of all the rotations is averged and the DC component is removed. (d) and (f) are the frequency responses of the kernels (c) and (e), respectively

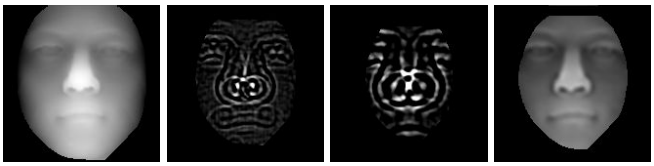


Figure 3: (a) is a facial range image. (b) and (c) are blob-like image which are computed by convolving the range image (a) with the kernels shown in Fig. 2.c and Fig. 2.e, respectively. The interest-points are extracted from the peaks in (b) and (c) which are easily detectable compared to the LoG in (e).

Unfortunately, the kernel has revealed to be sensitive to rotations around z axis (roll rotation). Although, some peaks in the blob-like image are repeatable some other points can go wrong. To circumvent this problem, we take the real part of r and rotate it around the central point from 0° to 360° in steps of 1°. At each step the kernel was bi-linearly interpolated at the integral values of x and y indices. The 360° interpolated kernels were then averaged and the DC value was subtracted. The resulting kernel h has complete invariance to roll rotations and can produce suitable blob-like patterns (see Fig. 2 and Fig. 3). The frequency response of h shows that it alternatively passes and suppresses consecutive rings of frequencies (their phase also alternates from 0° to 180°) that overlap with the initially selected frequencies (see Fig. 2.d and 2.f). Note that the resulting filter differs from LoG filters which are widely used for 2D interest-point detection. It has more oscillations and less weight at the center. Our filters can generate patterns with much more prominent peaks (see Fig. 3).

Four kernels of different sizes h_1 (21mm$\times$21mm), h_2 (19mm$\times$19mm), h_3 (31mm$\times$31mm) and h_4 (29mm$\times$29mm) were computed to produce two independent representing blob-like images (one image from h_1 and validated by h_2 and the other one from h_3 and validated by h_4). The $\mathscr{F}$ frequencies which were used to extract the kernels are the highest 3×3 adjacent frequencies in the LB band (the band of frequencies which are low horizontally and vertically) namely, the $3rd$, $4th$ and $5th$ frequencies in both horizontal and vertical directions. These frequencies are formally described in Eqn. 3

$$\mathscr{F} = \{(mf_1, nf_2) \in \mathscr{F} \mid 3 \leq m \leq 5 \text{ and } 3 \leq n \leq 5\} \tag{3}$$

There are no sufficient frequencies with high energy in the LB band for another non-

overlapping 3×3 window to produce another independent representing image using a kernel of the same size. The aforementioned kernel sizes of h_1 and h_3 are chosen so that using the same frequency window, their frequency responses are non-overlapping. The frequencies of the first kernel h_1 range from $\frac{3}{21}$ to $\frac{5}{21}$ K cycle/m (horizontally and vertically) but in the case of h_3 the range is $\frac{3}{31}$ to $\frac{5}{31}$ K cycle/m.

The kernel h_2 is very close in size and frequency response to h_1 but h_3 is close to h_4. The two representing images are computed using h_1 and h_3 and the other two kernels are used to increase the reliability of the detected interest-points. The peaks of the patterns in the first and the second representing images are validated using the patterns generated by h_2 and h_4, respectively. If the peaks of the patterns are still detectable at the same locations or within a small distance from their actual locations in the representing images, they are deemed reliable and considered as interest-points (their locations are less sensitive to minor changes in the spatial frequencies). Note that kernels of different sizes (scales) are designed to produce different patterns. Hence, the approach of maxima across a large-range of scales which is used in the SIFT [6] for scale-invariance (not required for 3D) is not applicable here.

3.2 Steps of the Proposed Approach

The approach combines interest-points from the two representing blob-like images as follows

1. The range image is convolved with each one of the four kernels h_1, h_2, h_3 and h_4 to produce the representing and validating images R_1, V_1, R_2 and V_2, respectively.

2. The peaks of the patterns in the four blob-like images are found by detecting the pixels which are the largest in their 7×7mm neighborhoods.

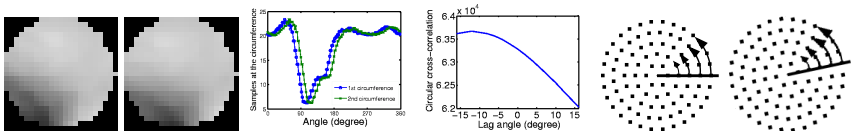
3. Each peak is assigned a strength measure s as in Eqn. 4. The peaks with low s are dropped.
$$s = R(x,y) \prod_{b(u,v)\in\mathscr{B}} R(x,y) - R(x+u,y+v) \tag{4}$$
where $\mathscr{B}$ is the set of border pixels of the 7×7 local neighborhood which have u and v offsets from the x and y location of the peak.

4. The set of peaks in R_1 that have corresponding peaks in V_1 within a distance of 2mm are called the first set of interest-points $\mathscr{I}_1$ and combined with the second interest-point set $\mathscr{I}_2$ which is extracted in a similar way from R_2 and validated by V_2 to form the total set of interest-points, $\mathscr{I}_t = \mathscr{I}_1 \cup \mathscr{I}_2$.

5. The non-maximum suppression (NMS) technique [10] is performed on the total set $\mathscr{I}_t$ to produce a filtered set $\mathscr{I}_f$. The points with maximal strength s suppress the inferior interest-points within a certain radius. See Section 5.1 for interest-points repeatability tests.

4 Face Recognition

For matching two facial range images, the interest-points in both images are detected as described in Section 3. Then, the local regions around the interest-points are matched against each other (Section 4.1). From the local region matches a similarity measure S between the matched facial images is computed.



(a) (b) (c) (d) (e) (f)

Figure 4: (a) and (b) are two local regions with a roll rotation between them. The curves in (c) are depth samples at their circumferences from which the roll angle is estimated using cross-correlation (d). The first region is sampled on concentric circles and the samples are vectorized staring from the angle 0 (the samples on the circles are appended in a depth vector starting from the largest circle to the smallest one) while the second local region is sampled and vectorized in accordance to the roll angle to achieve correspondence between the two sample vectors.

The shape of a human face deforms with expression, age and many other causes. Given such deformations, we may need to accept weak local region matches. The best match approach to the local regions produces some false matches resulting in a suboptimal total similarity measure. Instead of the best match approach to matching local regions, the local regions are putatively matched as described in Section 4.2.

4.1 Feature Extraction

The local region of each interest-point is sub-sampled on concentric circles as shown in Fig. 4.e and 4.f. This circular sampling facilitates the association between the depth values of the samples in the matched local regions. Under roll rotations we can achieve one-to-one correspondences between the depth values simply by vectorizing the sampling circles staring from an angle θ_s that equals the roll angle. It is worth mentioning that this circular sampling in the range image (only x and y coordinates are considered) is similar to finding the intersection between a sphere and a 3D surface [5, 2] (z coordinate is also considered). However, finding the relative positions of the samples to the interest-point position in case of circular sampling is more efficient computationally and can be computed offline in a look-up table while in the other case the intersection has to be computed online. On the other hand, the circular depth samples vary with pitch and yaw rotations. For small local regions (the sampling circle with largest radius is 15mm) and small pitch and yaw rotation angles the change is insignificant.

The vectorization starting angle θ_s provides invariance to roll rotations. To account for pitch and yaw rotations (surface orientations) of the matched local regions, the sampled depth values from the two local regions are linearly fitted to each other using an efficient least squares fitting technique. Then, the sum of absolute errors between the fitted depth values is used as a matching measure. The linear fitting gives invariance to pitch and yaw rotations. In addition, it mitigates the affect of difference in surface orientations on the circular sampling and minor errors in the locations of the detected interest-points. However, it does not have the flexibility to over-fit the depth values of the local regions as over-fitting dampens the error between dissimilar local regions leading to false matches.

The sum of absolute errors is computed from the local regions of two interest-points as follows:

1. The circumference of the first local region is sampled in steps of 1° and unfolded in

the counter clock-wise direction starting from angle zero ($\theta_s = 0$) into a sequence of depth values called p. In a similar manner, the circumference of the second local region is sampled and unfolded into q.

2. The roll rotation angle γ between the two local regions is found using the circular cross-correlation C between p and q as given in Eqn. 5

$$C(m) = \sum_{n=0}^{N-1} p_s q_n \quad \text{where} \quad s = \begin{cases} n+m & \text{for} \quad 0 \le n+m < N \\ N+n+m & \text{for} \quad n+m < 0 \\ n+m-N & \text{for} \quad n+m \ge N \end{cases} \tag{5}$$

There is no zero padding to the front or the end of the sequence p as in ordinary cross-correlation. Instead, the elements of the sequence are shifted from one end and inserted into the other end. The correlation sequence C is only computed over the range of lags m from -8 to 8 sampling angle steps. The roll angle is computed from the lag that yields maximum cross-correlation m_{max} by multiplying by minus the sampling angle step, $\gamma = -m_{max}$ (giving invariance within $\pm 8^\circ$, see Fig. 4.c and 4.d for an illustration example).

3. The first local region is sub-sampled on five concentric circles (see Fig. 4.e and 4.f). The radii of the circles are 15, 12, 9, 6, and 3 mm and the number of samples on the circles are 30, 25, 18, 12 and 6 respectively. The placement of the samples is based on the a starting angle $\theta_s = 0$. Each circle of samples is vectorized starting from the angle γ and appended to a vector $\mathbf{z_1}$, starting from the largest to the smallest circle. Similarly, the second local region is sub-sampled and vectorized into $\mathbf{z_2}$ but with a starting angle $\theta_s = \gamma$.

4. The depth vectors $\mathbf{z}_1$ and $\mathbf{z}_2$ are linearly fitted to each other. First, the depth at the center of the first local region (interest-point) z_{c1} is subtracted from $\mathbf{z}_2$, $\mathbf{z}'_2 = \mathbf{z}_2 - z_{c1}$. Then we adjust $\mathbf{z}'_2$ by adding a plane so that it fits $\mathbf{z}_1$ as in Eqn. 6.

$$\mathbf{z}''_2 = \mathbf{z}'_2 + \mathbf{L}_2 [u\, v]^\top \tag{6}$$

where $\mathbf{L}_2$ is a matrix of two columns: the first one is the vectorization of the x coordinates of the samples and the second one is the vectorization of their y coordinates. The u and v parameters that define the adjusting plane is computed by the following Eqn. 7.

$$[u\, v]^\top = (\mathbf{L}_2^\top \mathbf{L}_2)^{-1} \mathbf{L}_2^\top (\mathbf{z}_1 - \mathbf{z}'_2) \tag{7}$$

5. Finally, the sum of the absolute errors e between $\mathbf{z}_1$ and $\mathbf{z}''_2$ is computed (Eqn. 8).

$$e = \sum_{n=1}^{n=N} |\mathbf{z}_1(n) - \mathbf{z}''_2(n)| \tag{8}$$

4.2 Matching a probe to Face Gallery

During the offline phase, the interest-points are detected in the gallery faces as described in Section 3.2. Then their local regions are sub-sampled and vectorized (Section 4.1). We also, compute a difficulty factor σ_i for every interest-point in the gallery face. These difficulty factors are used in weighting the contributions of the local region matches. In the situation when the local regions are flat, the local regions can strongly but falsely

match. Even if the match is correct, the information in a flat surface is low and may not be proportional to the matching measure. While in some other situations the local regions may be complex and rich in information but may have smaller matching measure. The difficulty factor is heuristically defined for an interest-point based on the sum of absolute errors of fitting the local region to a planar region (in other words, the local region is matched to a planar region as described in Section 4.1). The difficulty factor ($\sigma = \log(e) - 2$) is expected to be high for complex local regions. Then the interest-points with difficulty factors less than a threshold $\sigma_t = 250$ are dropped.

The local regions with difficulty factors $\sigma_i > \sigma_t$ in the gallery faces are putatively matched to the local regions in the probe face. Firstly, we randomly select a subset from them. Then this subset is matched against all the local regions in the probe face using the best match approach. After that, we take the matches with fitting error e less than a strong threshold $t_s = 85$ (more likely to be correct matches) and use the RANSAC algorithm [9] to find the rotation $\mathbf{R}$ and the translation $\mathbf{t}$ between the probe and the gallery faces. In case the number of the strong matches are not sufficiently large (in our case we used a threshold of seven strong matches), we include matches with medium strength to find $\mathbf{R}$ and $\mathbf{t}$ (the matches with e less than a medium threshold $t_m = 110$). If the number of the matches is still not enough (less than seven) or RANSAC has failed to find $\mathbf{R}$ and $\mathbf{t}$ that fit the matching points, the two faces are considered dissimilar and a value of zero is assigned to the similarity measure.

The rotation $\mathbf{R}$ and the translation $\mathbf{t}$ that relate the gallery face to the probe face are used to restrict the search scope for interest-points in the probe face corresponding to the remaining ones in the gallery face. $\mathbf{R}$ and $\mathbf{t}$ are applied to x, y and z coordinates of the interest-point to give an estimate of the location of the corresponding interest-point in the probe image. The local regions around the interest-points in the proximity of that location are matched to the local region around the interest-point in the gallery image. The best match among them is accepted as a match if it is less than a weak threshold $t_w = 165$.

The RANSAC algorithm [9] can robustly fit a model (in our case, the rotation $\mathbf{R}$ and the translation $\mathbf{t}$) to data (the x, y and z coordinates of the matching interest-points) in the presence of outliers (false matches). From four randomly selected matches, $\mathbf{R}$ and $\mathbf{t}$ are computed (four is the minimum number from which $\mathbf{R}$ and $\mathbf{t}$ can be computed in a least square fashion [11]). The mean location of the four interest-points in the gallery face $\mathbf{m}_g$ is subtracted from their locations. Then their x, y and z coordinates are stored in the rows of a 3×4 matrix $\mathbf{P}_g$ and similarly the matrix $\mathbf{P}_p$ is computed from their corresponding interest-points in the probe face. A matrix $\mathbf{A}$ that transforms $\mathbf{P}_g$ to $\mathbf{P_p}$ is found, $\mathbf{A} = \mathbf{P}_p\mathbf{P}_g^\top(\mathbf{P}_g\mathbf{P}_g^\top)^{-1}$. The rotation matrix $\mathbf{R}$ is the nearest orthonormal matrix to $\mathbf{A}$. The singular value decomposition of $\mathbf{A}$ is computed, $\mathbf{A} = \mathbf{USV}^\top$. The rotation matrix $\mathbf{R}$ is computed, $\mathbf{R} = \mathbf{UDV}^\top$, where the matrix $\mathbf{D}$ is diagonal and the elements on the diagonal are $\{1, 1, 1/\mathbf{det}(\mathbf{UV}^\top)\}$. Then the translation is found $\mathbf{t} = \mathbf{m}_p - \mathbf{Rm}_g$. Then RANSAC finds the number of matches that fit $\mathbf{R}$ and $\mathbf{t}$ (the matches with fitness function $f = |\mathbf{Rp}_g + \mathbf{t} - \mathbf{p}_p|$ less than a distance threshold D_t, where $\mathbf{p}_g$ and $\mathbf{p}_p$ are the locations of the interest-point in the gallery image and the matching interest-point in the probe image respectively). If the number of matches that fit $\mathbf{R}$ and $\mathbf{t}$ is high (more than a percentage threshold) they are accepted as the fitting model. Otherwise RANSAC iterates until a number of trials is exhausted and takes the best model fitting the data.

The similarity measure S is extracted from the local region matches as given in Eqn.9. The weights of the matches in S depend on their strengths and difficulty factors σ_i. $\mathscr{W}$,

$\mathscr{M}$ and $\mathscr{S}$ are the sets of weak, medium and strong matches, respectively. t_w and t_m are the weak and the medium thresholds and e_i is the fitting error.

$$S = \sum_{e_i \in \mathscr{W} \cup \mathscr{M} \cup \mathscr{S}} |(t_w - e_i)| \sigma_i + \sum_{e_i \in \mathscr{M} \cup \mathscr{S}} (t_m - e_i)^2 \sigma_i \tag{9}$$

5 Experiments and Results

5.1 Repeatability Test of Interest-points

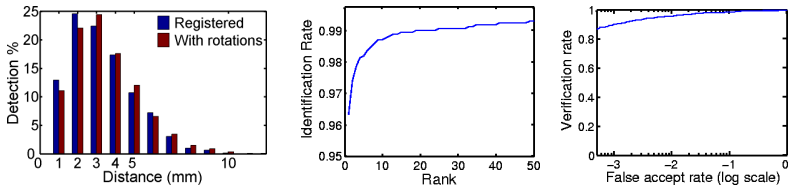
To test the repeatability and accuracy of the interest points, the approach was applied on a number of facial range images under neutral expression of many subjects. The detected interest-points that belong to the same subject were registered to each other using the ICP algorithm [8]. At a time, the ICP (3D) was applied on two sets of interest-points from a pair of range images. For each point in the first set, the 2D Euclidean distance (z is dropped as the error in z is not relevant to pixel accuracy) to the nearest point in the second set was found (distance error). A small distance indicates that the interest-point was accurately detected in the second range image. Fig. 5.a shows a histogram of error distance computed from all the range images in the test set. It shows that 60% of the interest-points were detected within a distance of 3mm and about 88% of the points are within a distance of 5mm.

The performance of the approach with respect to rotations was tested. The underling pointclouds of one of the range images of each pair were rotated using the pitch, yaw and roll angles in the range of $\pm 15^\circ$. After that, the range image was recalculated as in Section 2 (without the spike removal part). Then, the same repeatability test was performed on the range images (see Fig. 5.a).

In comparison to the performance without rotations, some degradation can be noticed in the accuracy of interest-point detection. However, generally the performance did not degrade significantly. The interest-points which were detected within an error range of 0 to less than 1mm has decreased from 13% to 12%. Also, the interest-points which were detected in the range of 1 to less than 2mm has decresed from 24.5% to 22% (see Fig. 5.a for more ranges). Staring from the range of 2 to less than 3 and onward, the interest-point detection has shows generally an increase in the detection percentage rather than a decrease which is an indication that some of the missed points in the smaller ranges were detected in the less accurate ones. The overall detection percentage was about 87.5% within 5mm. This repeatablity is comparable to typical 2D interest-point detection approaches when applied on 2D images [12, 6].

5.2 Recognition Results

The face recognition approach (Section 4) was applied to the FRGC V2.0 dataset [7], the largest publicly available dataset. The number of subjects in the test data is 466. Our approach to interest-point based face recognition was applied to the near frontal view 3D facial pointclouds with neutral expression. The range images are extracted from the pointclouds as described in Section 4. Each subject is represented by a single range image. The remaining range images are matched against the 466 gallery images. Fig. 5 shows the recognition performance of 1765 probes. The first rank recognition is 96.3% and increases to about 99% at the tenth rank (Fig. 5.b). The verification rate is about 90% at 0.1% FAR (Fig. 5.c).



(a) (b) (c)

Figure 5: (a) A histogram showing the repeatability of the detected interest-points in registered range images (no rotations) and in rotated images in the range of $\pm 15^{\circ}$. The width of the histogram bins is 1mm. (b) and (c) are the Cumulative Matching Characteristics (CMC) and the Receiver Operational Characteristics (ROC) showing face recognition performance.

6 Conclusions

An approach to interest-point detection in range images is presented and applied to facial range images. Experiments showed that the proposed technique can detect interest-points in facial range images with 88% repeatability at an accuracy of less than 5mm. We also presented novel algorithms for local feature extraction and feature matching for 3D face recognition. The proposed algorithms extract features around the interest points on a 3D face and match them for recognition under small pose variations. Experiments were performed on the FRGC v2 dataset and a rank one recognition rate of 96.3% was achieved.

References

[1] A. Mian, M. Bennamoun and R. Owens. *Region-based Matching for Robust 3D Face Recognition* . Proc. of BMVC, 2005.

[2] C. Chua and R. Jarvis. *Point signatures: A new representation for 3d object recognition*. IJCV, 1997.

[3] A. Moreno, A. Sanchez, J. Fco, V. Fco and J. Diaz . *Face recognition using 3D surface-extracted descriptors*. IMVIP, 2003.

[4] Y. Wang, C. Chua, Y. Ho and Y Ren. *Integrated 2D and 3D images for face recognition*. IEEE IAP, 2001.

[5] I. Mpiperis, S. Malasiotis, and M Strintzis. *3D Face Recognition by Point Signatures and Iso-contours*. Proc. of SPPRA , 2007.

[6] D. Lowe. *Distinctive Image Features from Scale-Invariant Keypoints* . IJCV, 2004.

[7] P. Phillips, P. 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.

[8] Y. Chen and G. Medioni. *Object modeling by registration of multiple range images*, IEEE PAMI, 1991.

[9] M. Fischler and R. Bolles. *Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography*. Comm. of the ACM, 1981.

[10] D. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice-Hall, 2003.

[11] S. Umeyama. *Least-squares estimation of transformation parameters between two-point patterns*. IEEE PAMI, 1991.

[12] M. Brown, R. Szeliski and S. Winder. *Multi-Image Matching using Multi-scale Oriented Patches*. IEEE CVPR, 2005.

Towards Automated Visual Assessment of Progress in Construction Projects

Timothy C. Lukins *
School of the Built Environment
Heriot-Watt University
Edinburgh, EH14 4AS, UK

Emanuele Trucco †
School of Computing
University of Dundee
Dundee, DD1 4HN, UK

Abstract

Current assessment of progress in construction projects is a manual task that is often infrequent and error prone. Images of sites are extremely cluttered and rife with shadows, occlusions, equipment, and people - making them extremely hard to analyse. We present a first prototype system capable of detecting changes on a building site observed by a fixed camera, and classifying such changes as either actual structural events, or as unrelated. We exploit a prior building model to align camera and scene, thus identifying image regions where building components are expected to appear. This then enables us to home in on significant change events and verify the actual presence of a particular type of component. We place our approach within an emerging paradigm for integration in the construction industry, and highlight the benefits of automated image based feedback.

1 Introduction

1.1 Context and Motivation

We propose a system capable of automatically detecting changes on a building site observed by a fixed camera, and to identify such changes as parts of the construction plan. Within the construction industry, the pressure is to always seek to deliver on time and on budget. Regular progress monitoring on construction sites is a fundamental part of project management required to achieve this. It allows payments to contractors to be paid periodically on the basis of the amount of work completed, and to respond to the many unexpected events affecting work (e.g., weather, supplies, accidents). Monitoring is currently carried out by independent surveyors, who inspect a site to check advancements against project schedule. However, these traditional approaches are often infrequent, subjective and manually intensive,

Recent trends in construction aim to represent every aspect of a whole project in a single, integrated Building Information Model (BIM). A BIM can include various aspects of a project, giving rise to "*nD*-systems" [1], containing not only the structural,

*t.lukins@hw.ac.uk
†manueltrucco@computing.dundee.ac.uk

geometric and material properties captured by traditional architectural designs but also schedule, health and safety, and cost information [8]. The BIM can furthermore be based on assemblages of standardised *components*, organised in turn into a hierarchy of *work-packages* [11]. The latter identify manageable and distinct project stages, and are defined by project-specific criteria, e.g., contractor, site location, or structure. Such a model, coupled with a suitable, interactive GUI, provides managers with complete visibility over a project in time. Importantly, the expected state at any time can be visualised, that is: what should have been built by a given date.

This ideally would be directly related to progress by keeping the model aligned with reality, i.e., the actual status of work on site. Unfortunately this proves very difficult. Currently, information collected from site surveys provide rather general assessments. Automated monitoring techniques have been proposed for tracking components, including bar-code labelling, RF tagging and embedded sensors, but their costs and practical aspects limit applicability [13]. Photographs of a site are taken as a matter of course but mainly just for recording purposes. Memon *et al.* [12] is one of the very few construction studies investigating imaging for progress assessment. The authors use a commercial package to recover a 3D model of the structures on site; these are then compared to the CAD model of the planned bulding. The process is however entirely manual and only works for simple structures, and is not so scalable to the complexities found of real construction sites.

1.2 Objectives and Contributions

We are partnering with the University of Salford to investigate the potential of computer vision as a progress assessment tool within a BIM framework. In general, this involves assessing automatically which elements of a building have been completed, by when, and consequently what stage of progress has been reached. Our main contribution is indeed an initial system detecting changes in image sequences acquired by a single, fixed camera placed in an uncontrolled location, and identifying changes as planned, i.e., expected events in the construction plan, or not. This system differs from other reported vision research by focusing on detecting close-range structural change, by bringing together a number of different techniques as follows.

Firstly, the system estimates the unknown position of the camera with respect to the building by line-based 3-D model matching similar to [5]. However, implementations of algorithms such as Simplex and SoftPOSIT failed to align given the complexity of the building model and the high clutter of real-site images. We achieved reasonably good results with a powerful evolutionary optimizer, *Particle Swarm Optimization* (PSO) [7].

Secondly, the aligned images in the sequence are normalized photometrically to assuage the effects of uninteresting lighting changes. This is followed by identification of regions in which changes have taken place since previous photographs. We use *Simoncelli filters* [16] to get good estimates of time derivatives. Currently we focus and retain only those change regions in which changes are actually scheduled within the BIM around the time the image was taken.

Thirdly, image changes do not *per se* imply that the expected component has been put in place. For this reason changes must be classified as significant (part of the building plan) or not. In other words, the component expected must be recognized in the target region; if not, the change is deemed uninteresting. Recognition is performed by an *AdaBoost* implementation with simple, Haar-like filters as per [18].

1.3 Related Work

The advantages of an integrated IT framework for the construction industry have been laid out by Bjork [3]. Trucco and Kaka [17] have then first discussed the potential of computer vision for progress assessment within such a framework. Their work presented a prototype vision system dealing with localisation of specific structures on a construction site using an iconic image matching approach. To our best knowledge, no papers exist in the computer vision literature on the automatic assessment of progress on building site as part of an integrated construction model. However, work exists on various related topics between the intersection of CV and buildings in general.

One topic that forms the focus of related research is in reconstructing building models from images. Traditionally, photogrammetry is the tool often employed by construction to measure a building from images. Many of these techniques underpin the core approaches in CV which focus on the multiple view recovery of structure. Building on previous developments in [2], Dick *et al.* [6] present a powerful framework in which a dictionary of common building elements and their learned distribution is fitted to observed geometry. Exploiting structural knowledge constrains the combinatorics of the problem. Cantzler and Fisher [4] presented an algorithm exploiting geometric constraints to improve the quality of automatically reconstructed building models. Schindler *et al.* [15] adopt similar techniques in the recent trend for "urban modelling" of entire cities from video as a structure from motion problem. They use the vanishing points to categorise and match the linear features often found on buildings. Notice that all this work is concerned with reconstruction, not with measuring building progress, and that the final models do not generally have the level of detail to accurately measure individual elements of the structure.

A second topic is quality assurance, e.g., tolerance verification, a typical task of photogrammetry [19]. This is a research area that is also founded on interpreting 3D data. For example, Gordon *et al.* [9] present a complete system to match scans of buildings to original 3D plans, in order to find and highlight elements deviating from prescribed tolerances. This approach to detecting defects can visually show and compare the "as-built" state to the original model. Scanning technology implies that the results can be very accurate, but there are issues related to the sheer volume of data to be interpreted.

The work above concerns change detection in close-range imagery, within 100 metres of the sensor. A great deal of other research exists on observation and verification of buildings using remote, long-distance imagery, including LIDAR and visible-spectrum images. For example, Huertas and Nevatia [10] report a system detecting specific changes in the overall structure of buildings. They exploit linear structure and seek to discard false edges created by shadows. Generalised 3D models are fitted to the result; and discrepancies used to justify identify changes.

Finally, we observe that progress assessment hinges on *change*, defined either as departure from an *a-priori* model [9, 10], or as difference between images taken at different times. An important realization is that the more prior knowledge of the nature of the change is modelled, the more the task becomes akin to model-based detection and location. Finding significant changes between images relates to the vast research area of *change detection*, especially for surveillance applications. For reasons of space, here we point the reader only to the excellent survey by Radke *et al.* [14].

In summary, our work fills a niche in that several vision algorithms have been reported for automatic model building, model improving and quality control, but nothing seems to exist on the automatic assessment of progress on building sites.

2 System Overview

The key problem is that construction sites offer some of the worst instances of a cluttered scene. The complexities of the structure, especially while it is being built, will result in certain ambiguities and self-occlusion. Furthermore, changes to the true structure over time, which indicate the progress we are interested in, may be surrounded by a large number of additional spurious events. These can broadly be classified as uninteresting changes, and include effects that are *environmental* (light and shadows, rain, snow, etc), *partial* (longer cumulative periods of work, such as pouring concrete), and *occasional* (people and equipment moving or staying still for a period of time).

The task of spotting correct changes can be split down into a number of modules:

- Localisation - to align a model of the building components to a sequence of images.
- Detection - to spot changes occurring over time within the region of components.
- Verification - to confirm after a change if a component is indeed present.

To make the task easier, we can engineer the camera set-up and leverage the knowledge we have from the construction model. For example, we assume that input is based on a fixed, single camera so that the initial camera-model localisation need only be performed once. Given the complete and accurate construction model, which includes CAD descriptions of the building elements, we can partition the image into regions where components should occur. Observing over time the intensity variations within these regions allows us to detect potential changes for verification.

3 Processing Modules

3.1 Camera to Model Alignment

This is effectively a model-based fitting approach in which we use a standard Canny edge detector with further post processing to join up strong neighbouring collinear lines in the image (within 0.1 radians similar direction and 30 pixels separation). The lines are represented as segments with end-point and angle (x, y, ρ) in a $\Re^3$ space. We represent the camera pose by 6 parameters, and the model is then rendered from this viewpoint and the same edge detection algorithms run to create a further set of model segments.

Non-linear optimisation of the camera pose then proceeds using a Particle Swarm based approach [7] to minimise the Sum of Squared Distances between the two sets of segments. PSO is a stochastic technique that iteratively searches across the multidimensional problem domain using a "swarm of particles" that are each guided by their own best solution, and by knowledge of the current global best for the entire swarm. Each particles velocity is thus governed by a weighing between these locations (which are updated on discovering new optima) and by inertia and randomising terms. This allows the fitting to initially proceed from a large number of possible solutions before rapidly converging towards the overall best global solution. This approach is well suited to a complex search space created by unrelated structures.

Additional constraints in the camera location (between 5 and 100 metres away, and not below the ground plane) also restricts the search. The output of this process is to produce a set of component regions, or *template masks*, created from the projection of the model onto the aligned image.

3.2 Component Change Detection

The biggest factor influencing false detection of changes in the image are the sudden effects of lighting. To remove as much as possible these variations, the image is first converted to grayscale, and the pixels within $n \times n$ sub-blocks are normalised to have zero mean and variance of one - as shown in Figure 1. Since we are considering such relatively small areas (in terms of the overall image) this approach is justified for removing a considerable amount of localised brightness and shadowing.

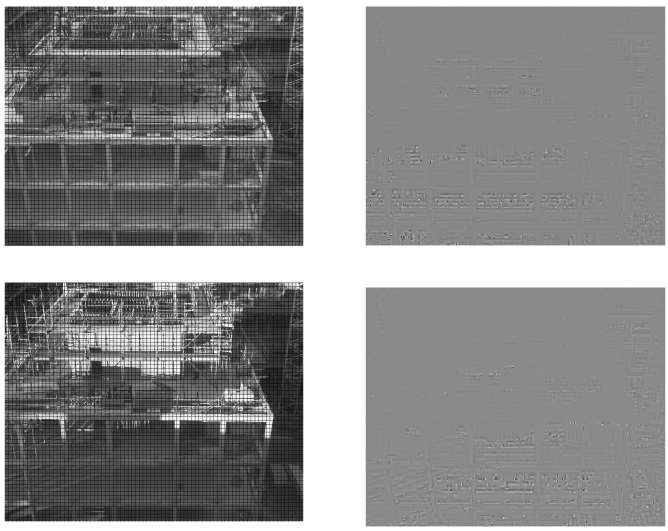


Figure 1: Gray-scale image *(top left)*, and same scene 20 minutes later with changed lighting conditions *(bottom left)*. Normalisation sub-blocks are then used to produce the respective corrected images shown retaining only low-frequency structure *(right)*.

We then wish to observe and mark sudden change occurring within particular regions with respect to time. For this, robust derivative estimation is achieved using a combination of $5 \times 5 \times 5$ two-stage low-pass (noise reduction) and high-pass (differentiate) 5-tap filters as developed by [16]. This is performed as a convolution first over the regions spatially using the smoothing kernel $[0.036, 0.249, 0.431, 0.249, 0.036]$ followed by the differentiation kernel $[-0.108, -0.283, 0.0, 0.283, 0.108]$ across time. We then calculate the mean for all the pixel temporal derivatives in the components template mask. A sensitivity threshold can then be set (as demonstrated experimentally in Section 4) to nominate peaks in the sequence as significant changes for further verification.

3.3 Verification of Components

We must now verify that detected changes are due to the appearance of scheduled components. We use a classifier constructed by the Adaboost algorithm based on a collection of "weak" classifiers [18]. We train the classifiers on a collection of 100 image samples

of the component (taken from other source images) which are further modified by varying degrees of rotation and intensity to mimic an increased training set of 1000 images. A further 2000 negative random samples from images of construction sites in which the component does not appear are also used. We employ the *Gentle* Adaboost algorithm based on a core set of non-rotated, linear Haar features to learn from this data.

The output of the classifier cascade when applied to an image is the location and scale of strong responses above a certain size as shown in Figure 2. This also highlights some of the Haar features learnt by the algorithm, showing the strong prevalence for "column like" edges and corners. Notice that in the output there are a number of false negatives, and also false positives in the incorrect matching to the smaller scaffolding poles at the top. However, what it important for the system is the presence of a column around the same time as change is detected. These locations can be directly compared to the distance from the centres of the component template mask in question. If verified, then the component is said to be present for the time of the originally detected peak.

Figure 2: Example output for column detection *(left)*, with Haar features used *(right)*.

4 Experimental Results

To illustrate this approach, we now describe a working implementation based on input from a real construction site. We use data from the construction of the new School of Informatics building at the University of Edinburgh. From this we consider a sequence of 1280×1024 images taken by a fixed camera every 20 minutes from 09:00 to 15:40 over 4 months (November '06 to February '07). We delete images taken over the weekend for a total of 1806 frames. As shown in Figure 3 this represents the construction of effectively three work packages of columns over increasing levels in the structure. Also shown is the 3D model which we we align on the basis of the ground floor components already in view. The resulting component template mask for 17 columns are further indicated in the figure. Notice we only consider those columns that are entirely in view, although it would be possible to also assess others which are not completely visible - especially when considering data from other cameras.

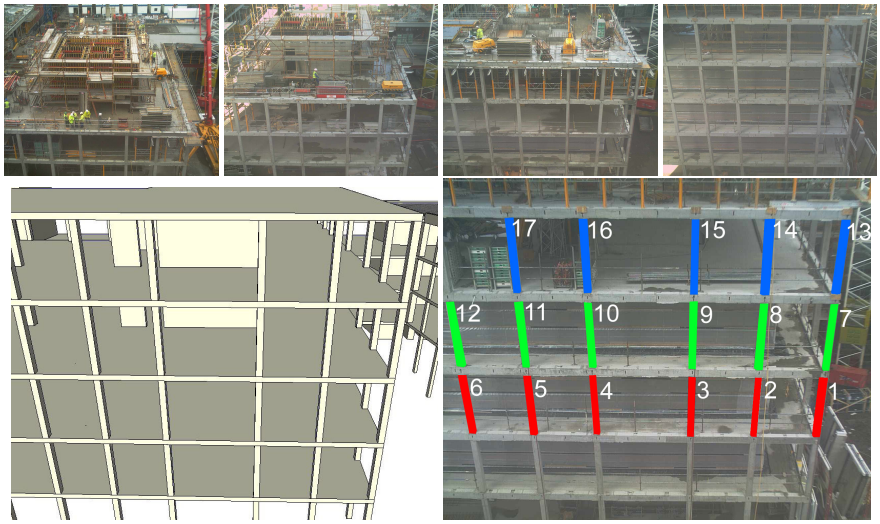


Figure 3: Input data sequence showing typical frames by month *(top)*, with aligned 3D model and resulting template masks for individual columns by work package *(bottom)*.

The alignment to perform this requires a degree of fine tuning to recover a good fit to the intrinsic parameters of the camera, particularly the focal length. This could be performed in advance by a calibration procedure. Optimisation of the pose proceeds based on the strong vertical and horizontal structure in the lower portion of the initial image. This is performed using 100 particles, taking 1000 iterations to converge to a reasonable solution. Following this, the template masks are generated from the projected outline of the final column positions. Each frame for a six-day period around expected construction time is loaded, converted to grayscale, and corrected for localised illumination using 64×64 blocks. The pixel values are then convolved with the smoothing and derivative filters. Taking the mean of the positive temporal derivative for each component template region reveals the plots as shown in Figure 4 for each period of construction, comparing the effects of illumination correction to the same process performed without.

The actual complexity and effectiveness of the graphs over these periods depends on the amount of overall work visible to the camera, and environmental factors during that time (i.e. periods of clear weather). For example, there is more activity and a large proportion of the rest of the site is still in view during the building of floor two. Notice that these can still cause numerous false negatives. However, illumination correction is able to remove many of these, especially the peaks caused by variation in the morning and evening (when the sun hits the site) as shown by the graphs on the right.

The performance of the final verification phase of the system is shown in Figure 5. ROC curves are established by comparing the correct timings of the columns against verification by Adaboost for detection thresholds ranging from 0.01 to 1.3 using both corrected and raw grayscale images. The best result is gained for corrected images with threshold 0.05, in which a total 16 of the 17 columns are correctly detected when they are actually installed.

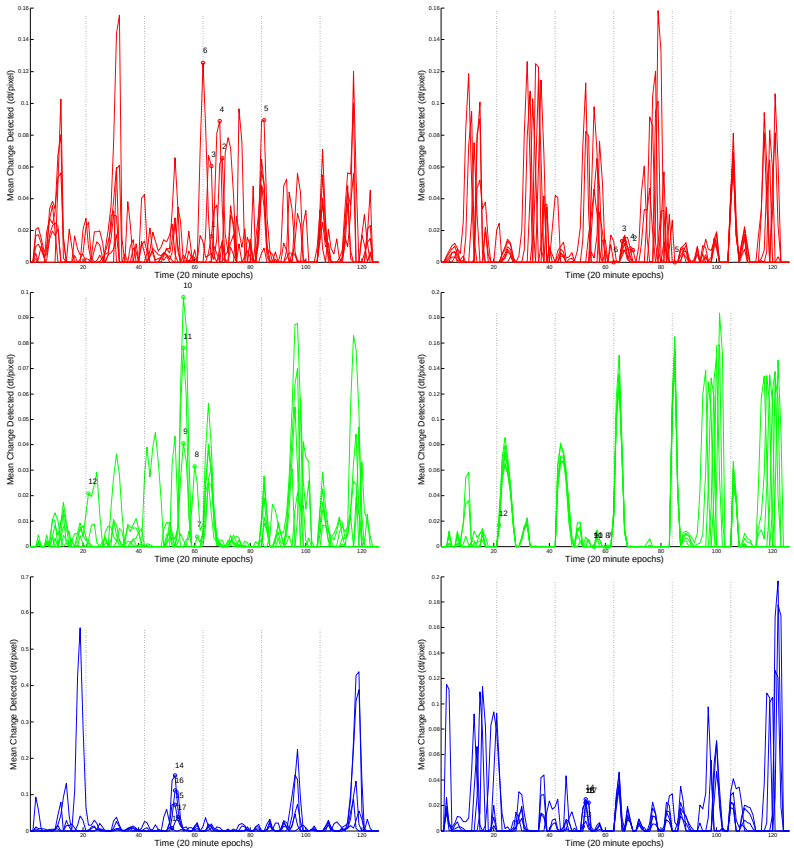


Figure 4: Change detection for 6 day period with illumination correction *(left)* and without *(right)*. Over the second floor *(top)*, third floor *(middle)*, and fourth floor *(bottom)* .

5 Conclusions

In this paper we have presented the application of techniques from computer vision to the task of reporting construction progress. We have shown how this is feasible by a combination of geometric model matching followed by statistical analysis of template mask regions to identify interesting changes for verification by an image based classifier. We have demonstrated how this approach can produce reasonable robust and reliable detection of key events during the construction process. The prospect of automated assessment represents huge potential for how large scale construction projects are managed.

However, a number of improvements are certainly possible. Firstly, in refining the model based fitting for image alignment, and to include input from other camera angles. This would enable us to handle occlusions in a more rigourous way (currently we ignore what we cannot see). Secondly, in making the detection algorithm more reliable by including texture and colour as indicators of change. In addition, more complex models of lighting could be used to remove illumination changes. Thirdly, in improving the verification classifier, especially for other types of component and differing materials.

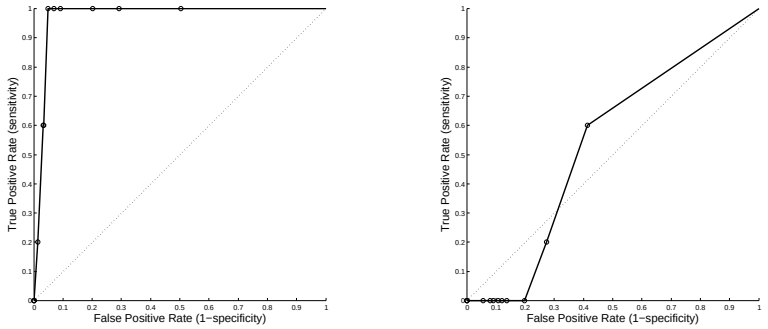


Figure 5: ROC curves for overall detection of 17 columns applying illumination correction *(left)* and without *(right)* with varying detection thresholds.

We are currently integrating this approach within an actual BIM framework in order to measure the true benefits for construction progress monitoring of an entire site. Even the simplest automatic confirmation of completion would enable further,more complex assessment (e.g. second floor activities require that the first floor be complete). In addition, individual work can be related to entire work-package progress, which will allow more meaningful and useful reports to be generated.

Acknowledgements

We thank Paolo Favaro, Yahaya Ibrahim and Ammar Kaka for useful discussions. We further thank the School of Informatics at the University of Edinburgh for the image sequence data, and Bennetts Associates (Architects) for the use of the simplified 3D building model. This research is funded by the UK EPSRC IMRC Outreach grant EP/C535200/1.

References

[1] G. Aouad, A. Lee, and S. Wu. nD modelling in construction: an integrated approach. *Journal of architectural engineering and design management*, 1(1), 2004.

[2] C. Baillard and A. Zisserman. A plane-sweep strategy for the 3D reconstruction of buildings from multiple images. In *19th ISPRS Congress and Exhibition*, Amsterdam, Netherlands, 2000.

[3] B.C. Bjork. Information technology in construction: domain definition and research issues. *International Journal of Computer Integrated Design and Construction*, 1(1):3–16, 1999.

[4] H. Cantzler, R. B. Fisher, and M. Devy. Improving architectural 3D reconstruction by plane and edge constraining. In *British Machine Vision Association Conference*, pages 43–52, Cardiff, UK, 2002.

[5] P. David and D. DeMenthon. Object recognition in high clutter images using line features. In *Internationa Conference in Computer Vision*, Beijing, China, 2005.

[6] A. R. Dick, P. H. S. Torr, and R. Cipolla. Modelling and interpretation of architecture from several images. *International Journal of Computer Vision*, 60(2):111–134, 2004.

[7] R. C. Eberhart and Y. H. Shi. Special issue on particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 8(3), 2004.

[8] C. Fu, G. Aouad, A. Lee, A. Marshall-Ponting, and S. Wu. IFC model viewer to support nD model application. *Automation in Construction*, 15(2):178–185, 2006.

[9] C. Gordon, F. Boukamp, D. Huber, E. Latimer, K. Park, and B. Akinci. Combining reality capture technologies for construction defect detection: A case study. In *9th EuropIA International Conference (EIA9)*, pages 99–108, Istanbul, Turkey, 2003.

[10] A. Huertas and R. Nevatia. Detecting changes in aerial views of man-made structures. *Image and Vision Computing*, 18(8):583–596, 2000.

[11] Y. Jung and S. Woo. Flexible work breakdown structure for integrated cost and schedule control. *Journal of Construction Engineering and Management*, 130(5):616–25, 2004.

[12] Z. A. Memon, M. Z. A. Majid, and M. Mustaffar. An automatic project progress monitoring model by integrating autocad and digital photos. In *ASCE International Conference on Computing in Civil Engineering*, Cancun, Mexico, 2005.

[13] R. Navon. Research in automated measurement of project performance indicators. *Automation in Construction*, 16(1):176–188, 2007.

[14] R.J. Radke, S. Andra, O. Al-Kofaha, and B. Roysam. Image change detection algorithms: a systematic survey. *IEEE Transactions on Image Processing*, 14(3):294–307, 2005.

[15] G. Schindler, P. Krishnamurthy, and F. Dellaert. Line-based structure from motion for urban environments. In *3rd International Symposium on 3D Data Processing, Visualization and Transmission*, Chapel Hill, USA, 2006.

[16] E.P. Simoncelli. Design of multi-dimensional derivative filters. In *IEEE International Conference on Image Processing*, volume 1, pages 790–793, 1994.

[17] E. Trucco and A. P. Kaka. A framework for automatic progress assessment on construction sites using computer vision. *International Journal of IT in Architecture, Engineering and Construction*, 2(2):147–164, May 2004.

[18] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 511–518, 2001.

[19] T. Werner, F. Schaffalitzky, and A. Zisserman. Automated architecture reconstruction from close-range photogrammetry. In *CIPA 2001 International Symposium: Surveying and Documentation of Historic Buildings, Monuments, Sites, Traditional and Modern Methods*, Potsdam, Germany, 2001.

Identifying Lens Distortions in Image Registration by Learning from Examples

Birgit Möller and Stefan Posch
Institute of Computer Science, Faculty of Natural Sciences III
Martin-Luther-University Halle-Wittenberg, 06099 Halle/Saale, Germany
{birgit.moeller,stefan.posch}@informatik.uni-halle.de

Abstract

Automatic quality assessment of image registration results is an important issue in image processing. Many applications strongly depend on accurate registration results, sometimes even requiring automatic self-recovery from registration failures. In doing so it is not sufficient just to detect registration errors. Also a distinction between different *error sources* is necessary, as each source has an individual impact on the final registration result and requires specific compensation strategies. We present a new approach to automatically identify *lens distortions* in image pairs, known to have a significant impact on registration. The key idea is to analyse registration residuals and to learn a model of spatial residual distributions typical for distorted images. Our approach relies on a new metric for registration quality assessment and implements a regression scheme based on SVMs for predicting distortions in unknown data. The potential of the approach is demonstrated by experimental results on synthetic as well as real image data.

1 Introduction

The analysis of similarities and differences between pairs of overlapping images, and the explicit modelling of image changes due to camera motion by (rigid) image transformations is a well-studied topic in computer vision. This so called *2D geometric image registration* yields the basis for a wide variety of applications, ranging from image sequence compression and mosaicing [9], up to the estimation of camera motion in space and geometric reconstruction of 3D scene structure [18]. The aim of registration is to determine an alignment of corresponding parts in partially overlapping images by estimating parameters of an appropriate model for the camera motion during image acquisition.

Within the last two decades large amounts of registration algorithms emerged [23]. On the one hand featureless approaches were proposed for parameter estimation (e.g., [13, 9]) that aim to minimise colour or intensity differences between images, e.g., using the well-known Mean Squared Error (MSE) as optimisation criterion. On the other hand, feature-based approaches are very common that build on an explicit detection of corresponding image points and, e.g., aim to minimise the reprojection error [7]. In general both classes of techniques allow for an accurate estimation of registration parameters. Accordingly, nowadays the basic task of 2D image registration is a quite mature field. But, nevertheless

there is still no guarantee for correct estimation results in any case, and registration errors as well as complete failures may still occur, and require appropriate treatment.

Despite promising advances in registration algorithms, an *automatic* and *objective assessment* of the final quality of a registration result is still an open issue. Although optimisation criteria used within the registration process itself in principal also provide a measure for the overall quality of the registration result, they often show a lack of local sensitivity (cf. [21, 14]). Also, the visual appearance of registered images as, e.g., observed by humans, does often not correlate very well to the registration quality as imposed by the optimisation function. Consequently, assessing the quality of registration results and especially determining the underlying error source in case of failure of the registration process is still a task usually left to the human user. This renders a fully automatic online image registration with automatic recovery from registration failures infeasible.

Geometric registration of two images may fail due to a wide variety of error sources, e.g., a lack of distinctive structures in the images, the choice of an inappropriate motion model or even simply image noise. One of the most important reasons, however, is given by *non-linear lens distortions* that are known to have a serious impact on the final result of a registration [8]. Accordingly, correcting images for these distortions prior to a geometric registration step is highly recommended. In literature several calibration algorithms are proposed that allow for robust camera calibration and lens distortion estimation. Among these approaches offline techniques based on calibration patterns [19, 1] turn out to form a quasi-standard with regard to accuracy. However, there are acquisition situations where these standard approaches are – if at all – difficult to apply, e.g., due to challenging environmental conditions (like underwater), or where a large flexibility is required as in case of continuously changing internal camera parameters due to zoom.

Image registration often forms a constitutional building block in image processing hierarchies, i.e., higher-level analysis modules strongly rely on accurate results of the registration stage. Due to this, in situations where a reliable calibration of the camera and an explicit compensation for non-linear lens distortions is not possible, it is at least advisable to thoroughly check the quality of any registration result. In particular, assessing the overall amount of distortions present in given images yields a valuable and often indispensable reliability measure for the outcomes of subsequent processing stages.

In this paper we present a new approach towards a fully automatic detection and quantification of lens distortions in registered images. The proposed method is part of an integrated approach for an automatic assessment of registration results [14]. As the final registration quality may be deteriorated by a wide range of potential error sources, any objective quality metric requires not only to detect remaining misalignments between two images, but also needs to distinguish between various underlying reasons. As these have individual impacts on the result they also request for individual compensation strategies.

Our proposed method for lens distortion detection builds on a new registration quality metric with high local sensitivity [14]. The key idea of the approach is to analyse spatial error distributions in so called *quality maps* that result from applying quality metrics to registered images. Lens distortions have shown to cause striking spatial patterns in these maps. Thus the quality maps serve as input for a machine learning approach that aims to predict the amount of lens distortions in image pairs by learning related effects from ground-truth examples. In this paper we extend an earlier case study dedicated to an initial categorisation of distortion artefacts into discrete classes [15] towards a continuous regression scheme. It allows for a direct and more distinctive prediction of distortions

using PCA-based dimension reduction and *support vector machines*.

The remainder of this paper is organised as follows. In Section 2 a brief overview of related work is provided. Section 3 summarises the basic principals of our metric for objective quality assessment. We predict distortions present in image pairs by performing a regression based on SVMs as outlined in Sec. 4. In Section 5 results of our approach are discussed. The paper finishes with a conclusion and an outlook on future work (Sec. 6).

2 Related Work

Although image registration algorithms play an important role in a wide variety of computer vision applications, the development of procedures for a fully automatic assessment of the quality of registration results has not yet been in the focus of the research community. Only few steps towards this direction were proposed [10], and work often focused on theoretically optimal boundaries of registration algorithms instead [3]. Consequently, objective metrics that would also allow for comparisons between different algorithms without requiring ground-truth data (which is often difficult to acquire) are not available, and quality assessment is usually done manually by humans.

In contrast to this lack of objective quality metrics with regard to image registration, within the field of *image quality assessment* over the years various metrics for measuring differences between images emerged [21]. These metrics usually aim at quantifying differences between images that show the same contents, however, may have undergone certain transformations or filtering operations, e.g., being compressed for data transmission purposes or degraded by various types of noise. Also, in virtual reality applications these metrics are common to compare artificially rendered scenes with real images.

Metrics in the field of image quality assessment usually exploit either visual image properties, emulating capabilities of the human eye (e.g., [12]), or structural properties of the images, e.g., local gradients [22], mutual information [16] or local image statistics of gray values [21]. Especially the second class of metrics is closely linked to an assessment of registration quality since image registration mainly aims at a structural alignment of images, i.e., minimising structural image differences. Accordingly, our metric builds on these approaches. However, as they have shown a lack of local sensitivity with regard to assessing registration results, we follow a pattern based approach for global analysis of image differences rather than applying common averaging schemes. This yields an increased local sensitivity indispensable in registration quality assessment.

Due to the lack of objective quality metrics in registration, also the undoubtfully serious impact of lens distortions in image registration has up to now mostly been evaluated qualitatively. E.g., in [8] the influence of lens distortions on accurately closing 360° panoramas is analysed, however, the results are presented in terms of visual maps to be interpreted manually by the human observer. This coincides with the fact, that even an objective assessment of the outcomes of stand-alone algorithms for recovering lens distortion coefficients is still an open question, if no ground-truth data is available.

As the effects of lens distortions and camera motion in image pairs are often tightly coupled, it is quite difficult to separate them uniquely from each other [11], as, e.g., aimed at by algorithms for an integrated recovery of camera motion and (radial) lens distortions [17, 6]. Hence, techniques to assess the amounts of distortions in image pairs as discussed in this paper are of significant interest, especially with regard to the goal of achieving high-quality image analysis results that rely on an accurate image registration.

3 A Metric for Registration Quality Assessment

The overall quality of a registration result is usually associated with the amount of pixel-wise differences between two registered images, remaining after alignment. Accordingly, analysing residuals between registered images yields a suitable starting point for any objective assessment. However, in doing so it has to be taken into account that image residuals not only emerge from registration failures, but may also be caused by other error sources, not directly related to the registration process itself.

We distinguish between *visual errors* and *structural errors* (Fig. 1). Visual errors are exclusively due to visual image differences like illumination changes or vignetting. Structural errors are caused by geometric misalignment of the images, i.e., due to a true registration failure.

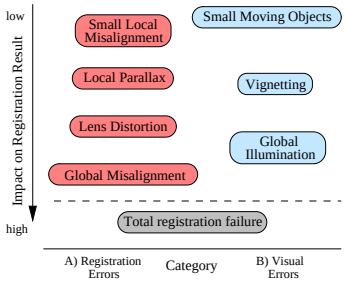


Figure 1: A taxonomy of image differences in registration (cf. [14]).

Our objective metric accounts for both classes of errors by thoroughly analysing local image structure and intensity differences. In Fig. 2 an overview of the two-step algorithm is provided. In the first stage three different pixel-wise quality criteria are calculated. Subsequently these are used as input in two independent analysis processes for detecting structural and visual differences between the images. Below the two stages of the algorithm are outlined. As for an analysis of lens distortions only structural differences are of interest, details about the visual error detection are omitted here, but can be found in [14].

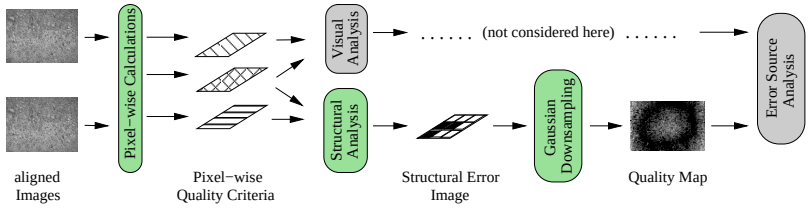


Figure 2: Overview of the metric for assessing the quality of a registration result.

3.1 Pixel-wise Quality Criteria

In the first stage three different quality criteria are calculated within the overlapping area of two registered images. These are the pixel-wise intensity difference, an *edge preservation map E* (cf. [22]), and finally a *structural risk map R* (cf. [5]) to separate structural and visual errors. Regarding structural error detection, only E and R are relevant.

Edge Preservation Map E

Image misalignment coincides with structural image differences. The edge preservation map exploits variations in the local gradient orientation between two images for detecting such defects, applying a perceptually motivated distance function:

$$E(x,y) = \frac{\Gamma_\alpha}{1+e^{k_\alpha(A(x,y)-\sigma_\alpha)}} \;,\; A(x,y) = 1 - \frac{|\alpha_1(x,y)-\alpha_2(x,y)|}{\pi/2} \;,\; \alpha_k = \tan^{-1}\left(\frac{s_k^y(x,y)}{s_k^x(x,y)}\right) \quad (1)$$

$s_k^x(x,y)$ and $s_k^y(x,y)$ are the results of the Sobel operator, and $\Gamma_\alpha = 0.9879$, $k_\alpha = -22$ and $\sigma_\alpha = 0.8$ are constants set according to the defaults suggested in [22].

Structural Risk Map R
At positions where the gradient magnitude is small in both images, structure is only weakly distinctive and the analysis of gradient orientations may lead to wrong conclusions. Accordingly, these positions are marked in the binary risk map to be excluded from structural analysis, and exclusively considered in visual error analysis:

$$R(x,y) = \begin{cases} 1, & \text{if } G_1(x,y) \leq \theta_G \wedge G_2(x,y) \leq \theta_G \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

G_1 and G_2 are the local gradient magnitudes in both images, and θ_G is a suitable threshold. Morphological dilation with a 3×3 square mask is applied to the risk map to also exclude pixels close to a homogeneous neighbourhood.

3.2 Global Assessment for Lens Distortion Analysis

For assessing the overall registration quality of an image pair, as outlined in our initial approach [14], a block-wise error pooling of edge preservation maps is performed, followed by a global voting procedure that yields four different error ratios for global error assessment. Regarding an identification and quantification of lens distortions as considered in this paper, however, the algorithm is slightly modified to better preserve local information.

Basically, the edge preservation map E is downsampled by a factor of eight, applying Gaussian masks for low pass filtering. In doing so, only pixels with distinctive gradient magnitudes in both images as indicated by the risk map R are considered. Figure 3 depicts some representative example *quality maps*, resulting from registering images degraded by varying amounts of non-linear distortions.

Figure 3: Example maps (contrast enhanced for visualisation), covering small (left) to big (right) distortion.

4 Predicting Lens Distortions with SVMs

The quality maps as calculated in the previous section reflect the overall registration quality in terms of their energy. However, with regard to lens distortion analysis, which is the main goal of this paper, not only the total amount of structural errors, but also their spatial distribution plays an important role. In particular, as can be seen in the example maps in Fig. 3, lens distortions cause striking spatial error patterns in the quality maps. The larger the degree of distortion within the registered images, the more pronounced are the radial symmetric patterns observed in related maps. Consequently, learning the relationship between the various characteristic error patterns and related amounts of radial distortion from ground-truth data should allow to quantify lens distortions in pairs of images.

To automatically quantify distortions according to spatial error distributions, an adequate measure for the degree of distortion in image pairs is required. Also, a suitable mapping function is necessary that relates the quality maps or accordant feature representations of these maps to the distortion measure. In our approach, we use a model-independent distortion measure (Sec. 4.1), and apply support vector machines to learn the transformation from the quality maps to this measure as a regression function (Sec. 4.2).

4.1 Characterising Lens Distortions

Radial lens distortions in image analysis are usually modelled using a radial-symmetric non-linear *polynomial mapping* between ideal points $\mathbf{p}$ in undistorted images and related distorted points $\mathbf{p}_d$ as observable from distorted images [7]:

$$\mathbf{p} = \mathbf{p}_0 + (1 + k_2 \cdot r^2 + k_4 \cdot r^4 + \ldots)(\mathbf{p}_d - \mathbf{p}_0), \tag{3}$$

where the k_i are called *distortion coefficients*, $\mathbf{p}_0$ denotes the center of radial distortion (which is often assumed to be identical to the image center), and $r = \| \mathbf{p}_d - \mathbf{p}_0 \|$ gives the Euclidean distance of point $\mathbf{p}_d$ to this center of distortion. Basically, the coefficients k_i of this model would yield a straight-forward measure for the distortion present in given images. Estimating a mapping function between spatial error distributions is in this case directly related to the reconstruction of parameters for a specific model of distortion.

However, since the direct reconstruction of coefficients k_i is sometimes ambiguous and we do not want to link our approach to a specific distortion model, we use a model-independent distortion metric Δ_{avg}. The distortion within an image is characterised by the average pixel offset caused by the applied distortion. Δ_{avg} is calculated from a subset of pixels located at equidistant positions on a ray through the center of distortion, which is sufficient to adequately characterise radial symmetric distortions as assumed here.

4.2 Learning Distortion Effects from Examples

The prediction of distortion from quality maps may be cast in the framework of function approximation. We use support vector machines (SVMs) in ε-regression mode [4] to achieve good generalisation abilities. The width of the insensitivity tube of the ε-insensitive loss function $|\xi|_\varepsilon$,

$$|\xi|_\varepsilon := \begin{cases} 0 & \text{if } |\xi| \leq \varepsilon \\ \varepsilon - |\xi| & \text{otherwise} \end{cases}, \tag{4}$$

is adjusted with the parameter ε. As usual, the parameter C denotes the tradeoff between the accuracy of the approximation on the training set and the capacity of the regression system. As kernels we use the a linear kernel, a polynomial kernel with degree 2 and a Radial Basis Function with γ controlling the width of the kernel.

In principle, the quality maps could directly be used as input for the SVM. However, as the maps usually have a size of about 60 to 80 pixels in width and 30 to 50 in height, this would result in a few thousands of input variables. This usually poses problems for generalisation, even for SVMs, and also for the computational load. Thus, we use PCA for feature extraction to project the quality maps into lower dimensional subspace. Subsequently each component of the resulting input vector is scaled to the interval $[0,1]$.

5 Experiments & Results

Given our objective metric for registration quality assessment and the model-independent distortion measure Δ_{avg} to quantify the amount of distortions present in images, the overall goal of our approach is to automatically identify lens distortions in pairs of registered images, and to assess their influence on the quality of the registration result from spatial residual distributions. To this end the SVMs are trained on quality maps generated by assessing registration results of image pairs with known distortion (Subsec. 5.1 and 5.2), and then applied to maps of registered synthetic as well as real test image pairs (Subsec. 5.3).

5.1 Sample Image Data

Learning a regression function requires a representative set of sample image pairs and related quality maps with known radial distortion. Unfortunately, real image data suffering from a large variety of different kinds and amounts of radial distortions, for which also a ground-truth calibration is available, is rare. Accordingly, we created ground-truth data from real image sequences with sufficient structure to allow for representative spatial error patterns. First, the raw images were corrected for radial lens distortions using a calibration pattern and the Matlab toolbox [1]. Subsequently pairs of images to be registered were selected and subjected to synthetic radial lens distortions of varying amount.

In detail, a set of 6.400 image pairs, 520×360 pixels in size, was selected from a real sequence showing a stony yard scanned by a camera moving in vertical direction in parallel to the ground. In Figure 4 an example image pair is depicted. The two images of each pair were randomly chosen from the sequence to ensure a reasonable variety of different offsets and image transformations in the data set. However, the offset in each pair was restricted to a maximum of 40 pixels to enable robust registration. Accordingly, given a moderate camera motion, the images of each pair were separated by up to 7 frames.

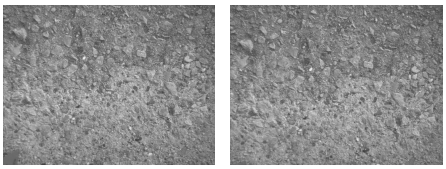


Figure 4: An exemplary image pair as used in training and test, showing a planar, stony yard.

To distort the images we applied varying radial distortion as defined in Eq. (3), assuming the distortion center in the image center. k_2 was sampled uniformly from the range $\left]0, 2.5 \cdot 10^{-6}\right]$ and k_4 from $\left]0, 2.5 \cdot 10^{-11}\right]$, giving values for Δ_{avg} between 0 and ≈ 30.

5.2 Experimental Setup and Training Phase

Once the sample set of distorted image pairs is given, each pair is registered by estimating a homography using the method proposed in [13]. Subsequently, the registration quality is assessed applying our metric, resulting in a quality map for each registered image pair of the sample set. They yield the base for training and testing the SVMs. 6000 quality maps were used in the training, while the remaining 400 formed the synthetic test set.

The first training stage consisted of determining a suitable feature subspace for reducing the dimensionality of the input data by PCA. To reduce the overall computational load only a subset of 2760 training samples was used in this step. In doing so the maximum number of relevant eigenvectors is bounded by the number of input images, restricting the size of the covariance matrix and simplifying the eigen analysis (details can be found in [20]). According to the eigenspectrum of the input data set we used between 10 and 60 eigenvectors (EVs) in our experiments to represent the feature space (see also Table 1).

All experiments were based on the *libSVM* [2] that supports easy training and testing as well as parameter optimisation. For each SVM initially appropriate parameters C, γ and ε (Subsec. 4.2) were determined by performing a grid search on the parameter space. A discrete sampling scheme with equidistant spacing in the logarithmic scale was applied to each of the parameters, and the resulting SVM-configurations were evaluated on the given training data subset using 5-fold cross-validation. Once the optimal SVM parameter set according to a minimal MSE on the training data was found for each SVM, they were trained on the full set of 6000 samples, and tested as outlined below.

#Eigenvectors (EVs)	SVM_{linear}	SVM_{poly2}	SVM_{RBF}
10	5.92	5.46	4.57
20	5.54	4.89	4.33
30	5.35	4.66	4.04
50	5.19	n.a.	**3.82**
60	5.13	n.a.	4.08

Table 1: RMSE of the SVM-regression approach on the set of 400 synthetic test images. "n.a." denotes experiments that were not carried out due to very high training times.

5.3 Results and Discussion

Different test runs were carried out using different dimensionalities for the feature vectors, and either linear kernels (SVM_{linear}), polynomial kernels with degree 2 (SVM_{poly2}), or kernels based on radial basis functions (SVM_{RBF}).

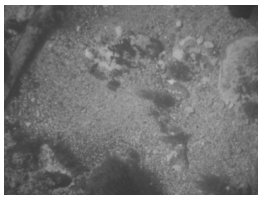


Figure 5: Example real test image acquired underwater.

Since the times for grid search and training of polynomial kernels turned out to be significantly higher than for the others, i.e., sometimes taking several days to terminate, for high dimensional data only linear and RBF kernels were applied. For testing two different data sets were used: i) the remaining 400 artificially distorted image pairs not being part of the training set, and ii) ten real image pairs acquired in a different domain, i.e., underwater. The latter ones were mainly tested to assess the generalisation capabilities of our approach.

In Table 1 the results of the test runs on the 400 synthetic images are summarised in terms of the roots of the Mean Squared Error (RMSE) for each run. In general RBF kernels yield the best results, and a selection of 50 EVs appears to be favourable. Polynomial kernels clearly outperform linear ones, but extraordinary high training times on high-dimensional data impair their practical relevance.

Pair	$\hat{\Delta}_{avg}$
0	16.1
1	14.0
2	13.7
3	20.0
4	19.9
5	18.2
6	17.0
7	16.4
8	15.8
9	17.6

Table 2: $\hat{\Delta}_{avg}$ for the set of 10 real image pairs.

The SVM that performed best on the synthetic test data, i.e.. using the RBF kernel with 50D vectors, was in the second test phase used to predict distortions $\hat{\Delta}_{avg}$ in unknown image pairs from a different domain, and also acquired with another autofocus camera (Fig. 5). Note that the focus was nearly constant during image acquisition according to camera motion and scene structure. As no ground-truth was available for this data set, the sequence was manually calibrated for lens distortions, based on a calibration pattern visible in some of the images. The estimated radial distortion relates to $k_2 \approx 1.5 \cdot 10^{-6}$, so that $\Delta_{avg} \approx 19$. In Table 2 the prediction results $\hat{\Delta}_{avg}$ for all ten real image pairs are summarised. In general the predicted values for the image pairs appear quite reasonable and meet the manual calibration result quite well, given a slight tendency to underestimate the amount of distortions. Although local image structure sometimes varies significantly within these images, the approach is able to interprete the overall residual distribution in a satisfying way, and to relate it to the known training samples for distortion prediction. This underlines the overall potential of our new technique, and particularly emphasises its generalisation capabilities with regard to unknown domains.

6 Conclusions

Objective registration quality assessment and an identification of underlying error sources is a key aspect in the development of fully automatic and robust image registration procedures. In this paper we have presented a new approach for automatically identifying lens distortions in image registration by analysing registration residuals and learning links between spatial error patterns and amounts of distortion from examples. Based on a new metric for objective quality assessment and a regression scheme using SVMs, our approach allows to predict the amounts of lens distortions in pairs of images, and to assess their impact on image registration. Tests on synthetic as well as real image data have shown the high potential of this technique, and also its generalisation capabilities with regard to various domains and acquisition devices. Future work will focus on a bigger variation of image transformations and offsets, including extended tests on real data. Finally, the approach will be integrated into the overall framework for automatic registration error analysis [14], yielding a fully automatic system for objective quality assessment.

Acknowledgements

This work was supported by a fellowship within the Postdoc-Programme of the German Academic Exchange Service (DAAD). The authors also would like to thank Dr. Rafael Garcia and the team of the Underwater Vision Lab at the University of Girona, Spain, for providing test images and related camera calibration data.

References

[1] J.-Y. Bouguet. Camera calibration toolbox for matlab, 2006. http://www.vision.caltech.edu/boujuetj/calib_doc/index.html.

[2] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[3] A. Criminisi, I. Reid, and A. Zisserman. A plane measuring device. *Image and Vision Computing*, 17(8):625–634, 1999.

[4] N. Cristianini and J. Shawe-Taylor. *An introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge Univ. Press, NY, USA, 2000.

[5] D. Farin and P.H.N. de With. Misregistration errors in change detection algorithms and how to avoid them. In *Int. Conf. on Image Proc.*, pages II:438–441, 2005.

[6] A. Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *Int. Conf. on Comp. Vis. and Patt. Recog.*, pages (1):125–132, 2001.

[7] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[8] S. Hsu and H. Sawhney. Influence of global constraints and lens distortion on pose and appearance recovery from a purely rotating camera. In *4th IEEE Works. on Appl. of Comp. Vision (WACV)*, page 154 ff., Princeton, NJ, USA, October 1998.

[9] M. Irani, P. Anandan, J. Bergen, R. Kumar, and S. Hsu. Efficient representations of video sequences and their applications. *Sign. Proc:Image Comm.*, 8:327–351, 1996.

[10] H.-S. Kim, H.-C. Kim, W.-K. Lee, and C.-H. Kim. Stitching reliability for estimating camera focal length in panoramic image mosaicing. In *Int. Conf. on Pattern Recognition*, pages 1:596–599, Barcelona, Spain, 2000.

[11] H. Li and R. Hartley. Self-calibration of radial lens distortion using two-view invariants. In *Proc. of ICCV, Omnivis Workshop*, Beijing, China, October 2005.

[12] J. Lubin. A visual discrimination model for imaging system design and evaluation. In E. Peli, editor, *Visual Models for Target Detection and Recognition*, pages 245–283. World Scientific, Singapore, 1995.

[13] S. Mann and R.W. Picard. Video orbits of the projective group: A new perspective on image mosaicing. Technical Report 338, MIT Media Laboratory Perceptual Computing Section, Boston, USA, 1996.

[14] B. Möller, R. Garcia, and S. Posch. Towards objective quality assessment of image registration results. In *Proc. of Int. Conf. on Computer Vision Theory and Applications (VISAPP '07)*, pages 233–240, Barcelona, Spain, 2007.

[15] B. Möller and S. Posch. Automatic analysis of lens distortions in image registration. In *Proc. of Workshop on Camera Calibration Methods for Computer Vision Systems (CCMVS '07), ICVS '07*, Bielefeld, Germany, March 2007.

[16] G. Qu, D. Zhang, and P. Yan. Information measure for performance of image fusion. *Electronics Letters*, 38(7):313–315, March 2002.

[17] H. Sawhney and R. Kumar. True multi-image alignment and its application to mosaicing and lens distortion correction. 21(3):235–243, 1999.

[18] P. H. S. Torr, A. W. Fitzgibbon, and A. Zisserman. The Problem of Degeneracy in Structure and Motion Recovery from Uncalibrated Image Sequences. *International Journal of Computer Vision*, 32(1):27–44, August 1999.

[19] R.Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, 1987.

[20] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[21] Z. Wang and A.C. Bovik. *Modern Image Quality Assessment*. Synth. Lectures on Image, Video & Multimedia Processing. Morgan & Claypool Publishers, 2006.

[22] C.S. Xydeas and V. Petrović. Objective image fusion performance measure. *Electr. Letters*, 36(4):308–309, Februar 2000.

[23] B. Zitová and J. Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, 2003.

Epsilon Stereo Pairs

Yuanyuan Ding Jingyi Yu
Department of Computer and Information Sciences
University of Delaware
Newark, DE 19716, USA
{ding,yu}@cis.udel.edu

Abstract

Human stereo vision works by fusing a pair of perspective images with a purely horizontal parallax. Recent developments suggest that very few varieties of multiperspective stereo pairs exist. In this paper, we introduce a new stereo model, which we call epsilon stereo pairs, for fusing a broader class of multiperspective images. An epsilon stereo pair consists of two images with a slight vertical parallax. We show many multiperspective camera pairs that do not satisfy the stereo constraint can still form epsilon stereo pairs. We then introduce a new ray-space warping algorithm to minimize stereo inconsistencies in an epsilon pair using multiperspective collineations. This makes epsilon stereo model a promising tool for synthesizing close-to-stereo fusions from many non-stereo pairs.

1 Introduction

A stereo pair consists of two images with a purely horizontal parallax. They provide important depth cues that are amenable to processing using computer vision algorithms [1, 4]. The classic pinhole and orthographic camera models have long served as the workhorse in capturing stereo pairs. However, recent developments have suggested that cameras that do not adhere to the pinhole structure may also form stereo pairs. For instance, a stereo panorama fuses two pushbroom images to synthesize a 360 degree depth perception [8]. We refer to these images as *multiperspective stereo pairs*.

Seitz [10] has classified all possible stereo pairs in terms of their epipolar geometry. Pajdla [6] independently obtained a similar result. They have shown that the epipolar geometry, if it exists, has to be a double ruled surfaces. Therefore, very few varieties of multiperspective stereo pairs exist. In this paper, we introduce a new framework, which we call epsilon stereo pairs, to model a much broader class of multiperspective images.

An epsilon stereo pair (ε-pair) consists of two images with a slight vertical parallax. We show many previous multiperspective cameras form valid epsilon stereo pairs even though they do not satisfy the stereo constraint. We use ε-pairs to model the recently proposed General Linear Cameras (GLC) [11]. We show both vertical and horizontal parallaxes can be directly derived from the GLC intrinsics.

We then present a new ray-space warping algorithm to minimize stereo inconsistencies in an ε-pair. Our approach is based on the multiperspective collineation theory [12], which describes the transformation between the images of a camera due to changes in

sampling and image plane selection. We show a proper collineation can effectively reduce stereo inconsistency in an ε-pair. Finally, we develop an automatic algorithm to find an optimal collineation via non-linear optimizations. This makes the epsilon stereo model a promising tool for synthesizing close-to-stereo fusions from many non-stereo pairs.

2 Previous Work

Classic pinhole cameras collect rays passing through a single point. Because of its simplicity, any oblique pair of pinhole images can be warped to have a purely horizontal parallax via projective transformations (homography) [4]. Recently, several researchers have proposed alternative multiperspective camera models, which capture rays originating from different points in space. These multiperspective cameras include pushbroom cameras [3], which collect rays along parallel planes from points swept along a linear trajectory, two-slit cameras [13], which collect all rays passing through two lines, and oblique cameras [7], in which each pair of rays are oblique.

Despite their incongruity of view, some multiperspective cameras can still form valid stereo pairs. Peleg et. al. [8] stitched the same column of images from a rotating pinhole camera to form a circular pushbroom. They then fused two oblique circular pushbrooms to synthesize a stereo panorama. Feldman et. al. proved that a pair of cross-slit cameras can have valid epipolar geometry if they share a slit or the slits intersect in four pairwise distinct points [2]. Seitz [10] and Pajdla [6] independently classified the space of all possible stereo pairs in terms of the epipolar geometry. Their work suggests that only three varieties of epipolar geometry exist: planes, hyperboloids, and hyperbolic-paraboloids, all corresponding to double ruled surfaces.

This paper focuses on how to fuse multiperspective camera pairs that do not have valid epipolar geometry. Such pairs may consist of two different cross-slit cameras, a cross-slit and a pushbroom, or two arbitrary multiperspective cameras. To simplify our analysis, we use the recently proposed General Linear Cameras (GLC) [11] to uniformly model these cameras.

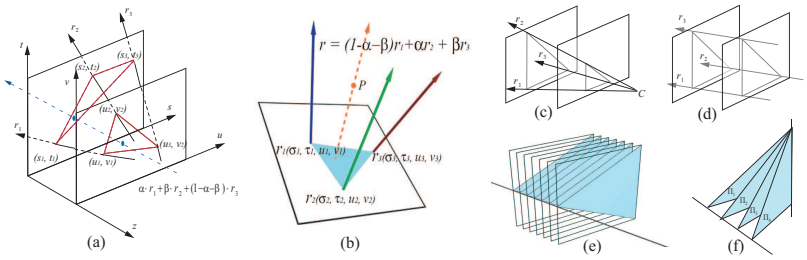


Figure 1: General Linear Cameras. (a) A GLC collects radiance along all possible affine combination of three rays. The rays are parameterized by their intersections with two parallel planes. (b) The GLC projection maps every 3D point P to a ray. The GLC model unifies many previous cameras, including the pinhole (c), the orthographic (d), the pushbroom (e), and the cross-slit (d).

2.1 General Linear Cameras

In the GLC framework, every ray is parameterized by its intersections with the two parallel planes, where $[s,t]$ is the intersection with the first and $[u,v]$ the second, as shown

in Figure 1(a). This parametrization is often called a two-plane parametrization (2PP) [5, 11]. Alternatively, we can reparameterize each ray by substituting $\sigma = s - u$ and $\tau = t - v$. In this paper, we will use this $[\sigma, \tau, u, v]$ parametrization to simplify our analysis. We also assume the default *uv* plane is at $z = 0$ and *st* plane at $z = 1$. Thus $[\sigma, \tau, 1]$ represents the direction of the ray.

A GLC is defined as the affine combination of three rays:

$$GLC = \{r : r = \alpha \cdot [\sigma_1, \tau_1, u_1, v_1] + \beta \cdot [\sigma_2, \tau_2, u_2, v_2] + (1 - \alpha - \beta) \cdot [\sigma_3, \tau_3, u_3, v_3], \forall \alpha, \beta\} \tag{1}$$

Many well-known multiperspective cameras, such as push-broom, cross-slit, linear oblique cameras are GLCs.

We further simply the GLC model by choosing three specific rays that have $[u, v]$ coordinates as $[0,0]$, $[1,0]$, and $[0,1]$ to form a canonical GLC:

$$r[\sigma, \tau, u, v] = (1 - \alpha - \beta) \cdot [\sigma_1, \tau_1, 0, 0] + \alpha \cdot [\sigma_2, \tau_2, 1, 0] + \beta \cdot [\sigma_3, \tau_3, 0, 1] \tag{2}$$

It is easy to see that $\alpha = u$ and $\beta = v$. Therefore, every pixel $[u, v]$ maps to a unique ray in the GLC.

Given a 3D point $P[x, y, z]$, the GLC projection maps P to pixel $[u, v]$ (see [12]) as:

$$u = \frac{\begin{vmatrix} z\sigma_1 & z\tau_1 & 1 \\ x & y & 1 \\ z\sigma_3 & 1 + z\tau_3 & 1 \end{vmatrix}}{Az^2 + Bz + C}, \quad v = \frac{\begin{vmatrix} z\sigma_1 & z\tau_1 & 1 \\ 1 + z\sigma_2 & z\tau_2 & 1 \\ x & y & 1 \end{vmatrix}}{Az^2 + Bz + C} \tag{3}$$

The denominator corresponds to the characteristic equation of the GLC:

$$Az^2 + Bz + C = 0 \tag{4}$$

where

$$A = \begin{vmatrix} \sigma_1 & \tau_1 & 1 \\ \sigma_2 & \tau_2 & 1 \\ \sigma_3 & \tau_3 & 1 \end{vmatrix}, B = \begin{vmatrix} \sigma_1 & v_1 & 1 \\ \sigma_2 & v_2 & 1 \\ \sigma_3 & v_3 & 1 \end{vmatrix} - \begin{vmatrix} \tau_1 & u_1 & 1 \\ \tau_2 & u_2 & 1 \\ \tau_3 & u_3 & 1 \end{vmatrix}, C = \begin{vmatrix} u_1 & v_1 & 1 \\ u_2 & v_2 & 1 \\ u_3 & v_3 & 1 \end{vmatrix} \tag{5}$$

The root $z_i (i = 1, 2)$ to equation (4) corresponds to a slit (line) on plane $z = z_i$ that all rays in the GLC will simultaneously pass through [11]. For instance, the cross-slit characteristic equation has two distinct roots since all rays simultaneously pass through two slits whereas an oblique camera has no solution.

Notice points on these slits will project to infinity on the image plane. To avoid this singularity, we assume the scene geometry $S(S^x, S^y, S^z)$ lie on the positive side of the image plane and is bounded by volume $[x_{min}, y_{min}, z_{min}] \times [x_{max}, y_{max}, z_{max}]$, where $z_{min} > max(z_1, z_2, 0)$.

3 Epsilon Stereo Pairs

A stereo pair consists of two images with a pure horizontal parallax, i.e., for every 3D point P, its images $[u, v]$ and $[u', v']$ in the two cameras must satisfy $v = v'$. We introduce a new stereo model, which we call epsilon stereo pairs, for fusing camera pairs that do not satisfy the stereo constraint.

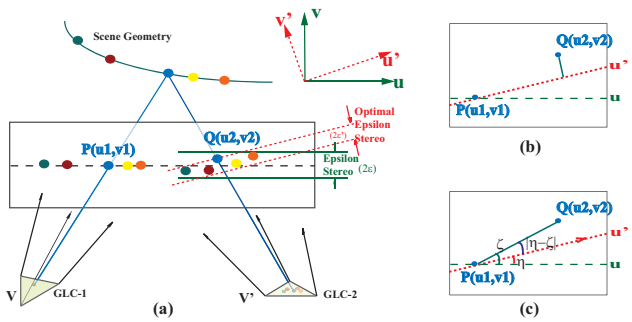


Figure 2: (a) An epsilon stereo pair consists of two images with a mostly horizontal parallax and a slight (ε vertical parallax. The vertical parallax is measured using a distance metric (b) or an angular metric (c). We can change the horizontal direction $\vec{d}$ to reduce ε.

We say that two views V and V' form an *epsilon stereo pair* if the following property holds:

The rays $V(u,v)$ and $V'(u',v')$ intersect only if $|v-v'| \leq \varepsilon$. The classical stereo constraint is a special case of the epsilon stereo model when $\varepsilon = 0$.

Any two such views are referred to as an ε-pair. In our analysis, we make the same assumption as [10], i.e., all views are *u*- and *v-continuous*. Furthermore, we only consider scene geometry visible in both views.

Physically, an ε-pair represents two views with a mostly horizontal parallax and with a slight (ε) vertical parallax. In an ε-pair, all pixels on a row in V correspond to pixels lying inside the ε band around the same row in V', as shown in Figure 2. When ε is small enough, our visual system can potentially fuse the two images as if they were stereo pairs, as shown in Figure 4.

3.1 Translation GLC Pairs

We show that any two GLCs form a valid ε-Pair. We first consider the pairs in which the second GLC is a translation of the first GLC along the image plane. Assume the first GLC is specified in its canonical form $[\sigma_1,\tau_1,0,0]$, $[\sigma_2,\tau_2,0,0]$, and $[\sigma_3,\tau_3,0,0]$, and the translation is specified as $[-t_x,-t_y,0]$.

The projection of a point $\dot{P}(x,y,z)$ in the first GLC can be computed using Equation (3). To project P to the second GLC, we can simply translate $\dot{P}$ by $[t_x,t_y,0]$. The vertical and the horizontal parallax of $\dot{P}$ can be computed as:

$$\Delta v = v - v' = -\frac{\begin{vmatrix} z\sigma_1 & z\tau_1 & 1 \\ 1+z\sigma_2 & z\tau_2 & 1 \\ t_x & t_y & 0 \end{vmatrix}}{Az^2+Bz+C} = \frac{z(t_y(\sigma_1-\sigma_2)+t_x(\tau_2-\tau_1))-t_y}{Az^2+Bz+C} \quad (6)$$

$$\Delta u = u - u' = -\frac{\begin{vmatrix} z\sigma_1 & z\tau_1 & 1 \\ t_x & t_y & 0 \\ z\sigma_3 & 1+z\tau_3 & 1 \end{vmatrix}}{Az^2+Bz+C} = \frac{z(t_y(\sigma_3-\sigma_1)+t_x(\tau_1-\tau_3))-t_x}{Az^2+Bz+C} \quad (7)$$

Notice, the x and y terms diminish in Δv and Δu. Thus, for a translation GLC pair, the vertical parallax Δv is a quadratic rational function of z. To show they form a valid epsilon stereo pair, we only need to prove $\Delta v(z)$ is bounded. We sketch the proof as follows.

Without loss of generality, we assume the characteristic equation (4) has coefficient $A > 0$ and has two roots z_1 and z_2 (if $A < 0$, we can analyze -$\Delta v(z)$ instead). Thus, equation (4) can be rewritten as $Az^2 + Bz + C = A(z - z_1)(z - z_2)$. We then replace it into (6) as:

$$\Delta v(z) = \frac{z(t_y(\sigma_1 - \sigma_2) + t_x(\tau_2 - \tau_1)) - t_y}{A(z - z_1)(z - z_2)} = \frac{\frac{1}{A}(t_y(\sigma_1 - \sigma_2) + t_x(\tau_2 - \tau_1))z - \frac{t_y}{A}}{(z - z_1)(z - z_2)} \tag{8}$$

Recall that the scene depth satisfies $z > max(z_1, z_2, 0)$, therefore, the denominator of (8) is positive and monotonically increasing. Next, we consider the numerator of equation (8).

If $\frac{1}{A}(t_y(\sigma_1 - \sigma_2) + t_x(\tau_2 - \tau_1)) < 0$, then the numerator is monotonically decreasing. In this case, $\Delta v(z)$ is monotonically decreasing, and its extrema can be computed as:

$$\Delta v(z)_{min} = \lim_{z \to \infty} \Delta v(z) = 0, \quad \Delta v(z)_{max} = \Delta v(z_{min}) \tag{9}$$

If $\frac{1}{A}(t_y(\sigma_1 - \sigma_2) + t_x(\tau_2 - \tau_1)) > 0$, we can rewrite the equation (8) as:

$$\Delta v(z) = \frac{\frac{1}{A}(t_y(\sigma_1 - \sigma_2) + t_x(\tau_2 - \tau_1)) - \frac{t_y}{Az}}{(1 - \frac{z_1}{z})(z - z_2)} \tag{10}$$

For $z > max(z_1, z_2)$, the denominator remains positive and monotonically increasing. For the numerator, when $t_y < 0$, it is monotonically decreasing. Thus, $\Delta v(z)$ is a monotonically decreasing function and its bound can be derived similarly to the derivation in (9). When $t_y > 0$, the numerator is a monotonically increasing. However it is bounded with maximum $\frac{1}{A}(t_y(\sigma_1 - \sigma_2) + t_x(\tau_2 - \tau_1))$ at $z = \infty$ and minimum at $z = z_{min}$. Thus, $\Delta v(z)$ is bounded, although it may not be monotonic w.r.t. z.

We have shown that Δv, in general, is bounded. In fact, in many cases, Δv is monotonically decreasing with respect to z. A similar derivation applies to Δu. Notice, the horizontal parallax, Δu, provides an important depth cue of the scene. The monotonicity in Δu is particularly important in stereo fusion. Our analysis indicates translation GLC pairs are not only ε-pairs but also well-suited for multiperspective stereo fusion if ε is small.

3.2 General GLC Pairs

Next, we consider a pair of two different GLCs. Assume the two GLCs are specified as $[\sigma_1, \tau_1, \sigma_2, \tau_2, \sigma_3, \tau_3]$, and $[\sigma'_1, \tau'_1, \sigma'_2, \tau'_2, \sigma'_3, \tau'_3]$. We have

$$\Delta v(x, y, z) = v - v' = \frac{\begin{vmatrix} z\sigma_1 & z\tau_1 & 1 \\ 1 + z\sigma_2 & z\tau_2 & 1 \\ x & y & 1 \end{vmatrix}}{Az^2 + Bz + C} - \frac{\begin{vmatrix} z\sigma'_1 & z\tau'_1 & 1 \\ 1 + z\sigma'_2 & z\tau'_2 & 1 \\ x & y & 1 \end{vmatrix}}{A'z^2 + B'z + C'} \tag{11}$$

where $(x, y, z) \in [x_{min}, y_{min}, z_{min}] \times [x_{max}, y_{max}, z_{max}]$

Since scene geometry lies beyond the slits of both GLCs, the denominator $Az^2 + Bz + C$ and $A'z^2 + B'z + C'$ cannot be zero. Furthermore Δv is linear in x and y and is bounded by $[x_{min}, x_{max}] \times [y_{min}, y_{max}]$. Therefore, we only need to consider Δv w.r.t z.

Recall that $v(z)$ and $v'(z)$ are both rational functions in z with quadratic numerators and denominators. Let z_1, z_2 be the two roots of the corresponding GLC characteristic equation for $v(z)$. We can rewrite the projection equation (3) as:

$$v(z) = \frac{\frac{1}{A}(\sigma_1\tau_2 - \sigma_2\tau_1) + \frac{1}{A}(y(\sigma_2 - \sigma_1) + x(\tau_1 - \tau_2) - \tau_1)\frac{1}{z} + \frac{y}{A}\frac{1}{z^2}}{(1 - \frac{z_1}{z})(1 - \frac{z_2}{z})} \tag{12}$$

Let $\kappa = \frac{1}{z}$,since scene points satisfy $z > max(z_1, z_2, z_{min})$, we have $\kappa \in (0, \frac{1}{max(z_1,z_2,z_{min})})$, and equation (12) transforms to:

$$v(\kappa) = \frac{\frac{1}{A}(\sigma_1\tau_2 - \sigma_2\tau_1) + \frac{1}{A}(y(\sigma_2 - \sigma_1) + x(\tau_1 - \tau_2) - \tau_1)\kappa^1 + \frac{y}{A}\kappa^2}{(1 - z_1\kappa)(1 - z_2\kappa)}$$

Thus, we have $(1 - z_1\kappa)(1 - z_2\kappa) \in ((1 - \frac{z_1}{max(z_1,z_2,z_{min})})(1 - \frac{z_2}{max(z_1,z_2,z_{min})}), 1)$. This proves $v(z)$ is bounded w.r.t to z. A similar proof applies to $v'(z)$, and hence, Δv is also bounded.

Notice, our analysis only applies to GLCs whose characteristic equations are quadratic, such as cross-slit and linear oblique cameras. The characteristic equation may degenerate to linear in the case of pushbroom cameras. However, in that case, the GLC projection is linear and rational, and a similar derivation can be applied.

3.3 Reducing Vertical Parallax

The vertical parallax in an ε-pair can be further reduced by using a different horizontal direction, as shown in Figure 2(a). Here, we present a simple and intuitive algorithm for finding the optimal horizontal direction for a GLC ε-pair. We assume that the two GLCs share the same image plane at $z = 0$. Our goal is to estimate the optimal horizontal direction $\vec{d}$ so that the average vertical parallax ε is minimal.

Assume the scene geometry is known, then, for each 3D point we compute projections in the ε pair as $P[u_1, v_1]$ and $Q[u_2, v_2]$ using equation (3). To measure the vertical parallax, we can compute the distance from Q to the scanline that passes through P with direction $\vec{d}$, as shown in Figure 2(b).

Assume the angle between $\vec{d}$ and the u-axis is η, this distance can be computed as:

$$d_{PQ} = |sin\eta(u_2 - u_1) - cos\eta(v_2 - v_1)| \tag{13}$$

The optimal η can be obtained by solving the following miminization problem:

$$\eta_{opt} = \min_{\eta} \int_{\Theta} |sin\eta(u_2 - u_1) - cos\eta(v_2 - v_1)|^2 dxdydz \tag{14}$$

Alternatively, we can measure the angular difference between $\vec{PQ}$ and $\vec{d}$ as shown in Figure 2(c). Let ζ denote the angle between $\vec{PQ}$ and u axis, we have:

$$\eta - \zeta = \eta - \arctan(\frac{v_2 - v_1}{u_2 - u_1}) \tag{15}$$

Hence the optimal horizontal direction can be computed as:

$$\eta_{opt} = \min_{\eta} \int_{\Theta} (\eta - \arctan(\frac{v_2 - v_1}{u_2 - u_1}))^2 dxdydz \tag{16}$$

4 Optimal Epsilon Stereo Pairs

When an ε-pair consists of cameras that capture significantly different rays, changing the horizontal direction on the image plane is insufficient for reducing the vertical parallax. In this section, we present a new ray-space warping algorithm based on multiperspective collineations.

4.1 Multiperspective Collineation

A multiperspective collineation describes the transformation between the images of the camera due to changes in sampling and image plane selection. This transformation is analogous to planar collineation (homography) in pinhole cameras.

A collineation can be specified by an image plane $\Pi[\dot{p}, \vec{d}_1, \vec{d}_2]$, where $\dot{p}$ specifies the origin of the plane and $\vec{d}_1$ and $\vec{d}_2$ specify the two spanning directions. For every ray r that has origin $\dot{o}$ and a direction $\vec{l}$, we intersect ray r with Π to compute its pixel coordinate $[i, j]$ as:

$$[o_x, o_y, o_z] + \lambda [l_x, l_y, l_z] \quad = \quad \dot{p} + \vec{d}_1 i + \vec{d}_2 j \tag{17}$$

Solving for i, j, and λ in (17) gives:

$$\begin{aligned} i &= \frac{(-l^z d_2^y + l^y d_2^z)(o^x - p^x) + (l^z d_2^x - l^x d_2^z)(o^y - p^y) + (-l^y d_2^x + l^x d_2^y)(o^z - p^z)}{\gamma} \\ j &= \frac{(l^z d_1^y - l^y d_1^z)(o^x - p^x) + (-l^z d_1^x + l^x d_1^z)(o^y - p^y) + (l^y d_1^x - l^x d_1^y)(o^z - p^z)}{\gamma} \end{aligned} \tag{18}$$

where

$$\gamma = \begin{vmatrix} d_1^x & d_2^x & -l^x \\ d_1^y & d_2^y & -l^y \\ d_1^z & d_2^z & -l^z \end{vmatrix} \tag{19}$$

For a canonical GLC, $[\dot{p}, \vec{d}_1, \vec{d}_2]$ are all linear functions of the pixel coordinate $[u, v]$. Therefore, the GLC collineation is, in general, a quadratic rational function.

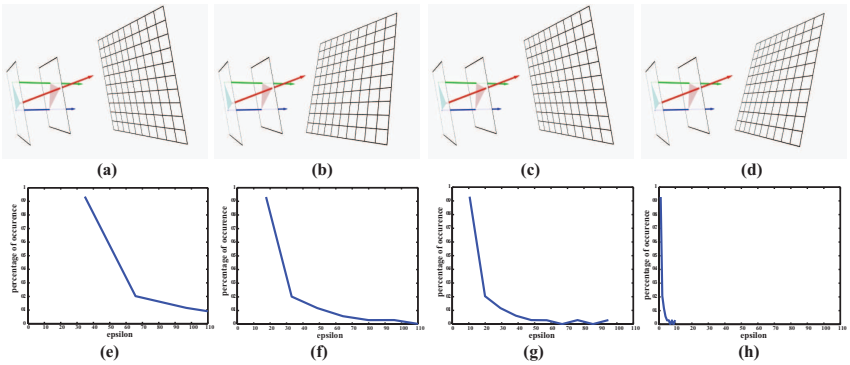


Figure 3: We use multiperspective collineations to reduce the vertical parallax between a pinhole and a cross-slit camera. Top row: We apply different collineations on the cross-slit camera. Bottom row: We plot the corresponding distribution of the vertical parallax. The vertical parallax is significantly reduced using collineation (d).

4.2 Automatic Collineation Optimization

Given an ε-pair, we use collineations to reduce stereo inconsistencies. In Figure 3, we apply different collineations to an ε-pair that consists of a pinhole camera and a cross-slit camera. We also densely sample the scene and plot the vertical parallax distributions

under each collineation using a histogram. The horizontal axis represents the range of vertical parallaxes and the vertical represents the occurrence percentage of the points. As shown in Figure 3(h), the vertical parallax of an ε-pair can be significantly reduced with a proper collineation.

We also present an automatic algorithm to robustly estimate the optimal collineation. Our algorithm starts with features matching between the two images. Recall that each GLC captures a different set of rays and exhibits unique distortions. Thus, matching the feature points between two GLCs can be difficult. In our implementation, we use Scale Invariant Feature Transform (SIFT) to preprocess the images. SIFT robustly handles image distortion and illumination variations and generates transformation-invariant features. We then perform global matching using SSD to find the potential matching pairs. To remove outliers, our algorithm uses RANSAC where we use projective transformations as its precondition.

Given feature correspondences $f_k(i_k, j_k), (k = 1, \ldots, n)$ in I and $f'_k(i'_k, j'_k), (k = 1, \ldots, n)$ in I', we compute the ray representation r'_k for each feature f'_k in I'. Our goal is to find the optimal collineation $[\dot{p}, \vec{d}_1, \vec{d}_2]$ that minimizes the vertical parallax between f_k and $Proj(r_k, \Pi)$, where $Proj(r'_k, \Pi)$ represents the projection of ray r'_k under collineation Π. As shown in equation (18). We formulate this optimization as a least squares problem:

$$\min_{\Pi} \sum_{r'_k \in feature raysin I'} D^2(Proj(r'_k, \Pi), f_k) \tag{20}$$

The distance metric $D^2(Proj(r'_k, \Pi), f_k)$ measures the vertical parallax between f_k and $Proj(r'_k, \Pi)$.

One way to compute D is to calculate the vertical distance between $Proj(r'_k, \Pi)$ and f_k. In Figure 4(b), we apply the vertical distance metric to fuse two cross-slit images. We use blue arrows to mark the displacements between the feature points in the final fusion. Although the vertical parallax is significantly reduced, many features pairs, such as the ones of the middle building, lie very close to each other and fail to provide correct depth cues.

We propose a different distance metric that simultaneously measures the vertical and the horizontal parallax for each feature pair. We estimate the intersection point of the corresponding two rays for each pair and assign an approximated horizontal disparity $disp_k$ that emulates how it would be seen in a perspective stereo pair. We formulate the new distance metric as:

$$D^2(Proj(r'_k), f_k) = (Proj(r'_k)^i - (f_k^i + disp_k))^2 + (Proj(r'_k)^j - f_k^j)^2 \tag{21}$$

Finally we use Levenberg-Marquardt to solve equation (20).

5 Results

We have performed experiments on various GLC pairs using our algorithm. We have modified the PovRay [9] ray tracer to accommodate different GLC types. In Figure 4, we fuse a pinhole image with a cross-slit image of a city scene. These two cameras cannot form epipolar geometry. To better illustrate our results, we color the pinhole image with green and the cross-slit with red and fuse the two images in an anaglyph. We apply

our automatic collineation algorithm to find the optimal collineations. We highlight the displacement between the corresponding features with blue arrows in the final anaglyph. The initial two images exhibit a severe vertical parallax (Figure 4(a)). We then compute the optimal collineation using the vertical distance. As a result, the vertical parallax is significantly reduced. However, many features lie too close to each other and do not provide valid depth cues (Figure 4(b)). We then use the disparity metric (21) to compute the best collineation. The resulting collineation produces a close-to-stereo pair with a minimal vertical parallax and a consistent horizontal parallax (Figure 4(c)).

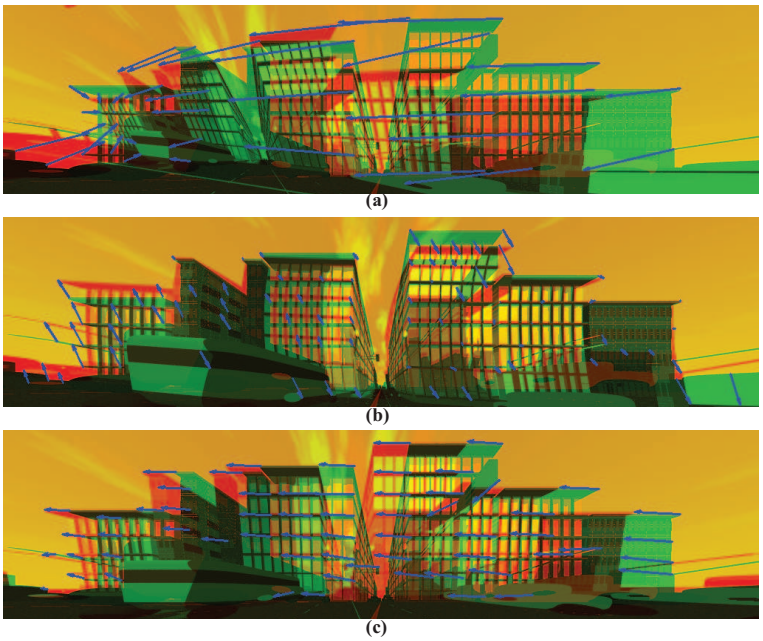


Figure 4: (a) An anaglyph of a pinhole (green) and a cross-slit (yellow). (b) An optimized ε-pair using the vertical parallax metric. Overall, the vertical parallax is reduce. However, the depth cue is also lost. (c) An optimized ε-pair using the disparity metric. The vertical parallax is reduced and the horizontal parallax is preserved.

We have also compared our collineation algorithm with the commonly used projective transformation. In Figure 5, we fuse a pinhole image (Figure 5(a)) and a pushbroom image with additional synthetic radial distortions (Figure 5(b)). Figure 5(c) computes the optimal projective transformation and Figure 5(d) computes the optimal collineation, both using the disparity metric (21). The optimal collineation result is less distorted and is more consistent with the pinhole image. This is because multiperspective collineation describes a much broader class of warping functions than the projective transformation.

6 Conclusion

In this paper, we have introduced a new framework, which we call epsilon stereo pairs, to model images captured by a broader class of multiperspective cameras. An epsilon

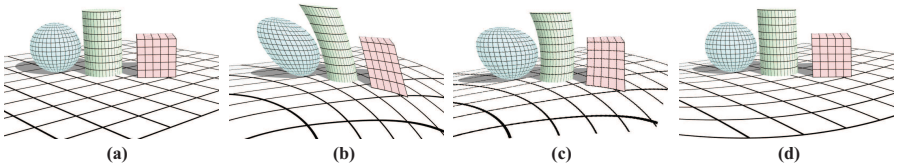


Figure 5: We compare collineation with the projection transformation on an ϵ-pair of a pinhole camera (a) and a pushbroom camera with synthetic radial distortions (b). (c) shows the the optimal projective transformation result. (d) shows our optimal collineation result.

stereo pair consists of two images that have a slight vertical parallax. We have shown many previous multiperspective images form epsilon stereo pairs, even though they do not satisfy the stereo constraint. We have also proposed a new ray-space warping algorithm based on multiperspective collineations to minimize the vertical parallax between two images. Experiments demonstrate that our epsilon stereo model provides a promising tool for synthesizing close-to-stereo fusions using non-stereo pairs.

Acknowledgement

This work has been supported by the National Science Foundation under grant NSF-MSPA-MCS-0625931.

References

[1] Bolles, R. C., H. H. Baker, and D. H. Marimont: Epipolar-Plane Image Analysis: An Approach to Determining Structure from Motion. International Journal of Computer Vision, Vol. 1 (1987).

[2] D. Feldman, T. Pajdla, and D. Weinshall: On the Epipolar Geometry of the Crossed-Slits Projection. In Proc. 9th Int. Conf. on Computer Vision (2003).

[3] R. Gupta and R.I. Hartley: Linear Pushbroom Cameras. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, no. 9 (1997) 963–975.

[4] R.I. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision. Cambridge Univ. Press, 2000.

[5] M. Levoy and P. Hanrahan: Light Field Rendering. Proc. ACM SIGGRAPH '96 31–42.

[6] T. Pajdla: Epipolar Geometry of some non-classical cameras. Proceedings of Computer Vision Winter Workshop, pp. 223-233, Slovenian Pattern Recognition Soceity.

[7] T. Pajdla: Stereo with Oblique Cameras. Int'l J. Computer Vision, vol. 47, nos. 1/2/3 (2002) 161–170.

[8] S. Peleg, M. Ben-Ezra, and Y. Pritch: Omnistereo: Panoramic Stereo Imaging. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 23, no. 3 (2001) 279–290.

[9] POV-Ray: The Persistence of Vision Raytracer. http://www.povray.org/

[10] S. M. Seitz: The Space of All Stereo Images. Proc. Int'l Conf. Computer Vision '01, vol. I (2001) 26–33.

[11] J. Yu and L. McMillan: General Linear Cameras. Proceedings of 8th European Conference on Computer Vision (2004), Volume 2, pp. 14-27.

[12] J. Yu and L. McMillan: Multiperspective Projection and Collineation. Proceedings of 10th Intl. Conferenc on Computer Vision (2005), pp. 580-587.

[13] A. Zomet, D. Feldman, S. Peleg, and D. Weinshall: Mosaicing New Views: The Crossed-Slits Projection. IEEE Trans. on PAMI (2003) 741–754.

Learning object classes from structure

Bai Xiao, Yi-Zhe Song and Peter M. Hall
Media Technology Research Centre
Department of Computer Science
University of Bath

Abstract

The problem of identifying the class of an object from its visual appearance has received significant attention recently. Most of the work to date is premised on photometric measures, often building codebooks made from interest regions. All of it has been tested only on photographs, so far as we know. Our approach differs in two significant ways. First, we do not build a codebook of interest regions but instead make use of a hierarchical description of an image based on a watershed transform. Root nodes in the hierarchy are putative objects to be classified. Second, we classify these putative objects using a vector of fixed length that represents the structure of the hierarchy below the node. This allows us to classify not just photographs, but also paintings and drawings of visual objects.

1 Introduction

The problem of identifying the class of an object from its visual appearance has received significant attention recently. This problem is to be differentiated from that of object segmentation; approximate segmentations of the object in question or even parts of the object are proving sufficient. The problem should also be differentiated from that of specific object identification, there are degrees of freedom open to classes of objects that are unavailable to specific objects.

There are many accounts of the problem of identifying visual classes, including active appearance models [4], pictorial structures [6] and tensor faces [20]. A category of solution of particular interest here acquires the statistics of appearance and relative spatial distribution of local features, in a semi-unsupervised environment (training assumes only that images depict an object known to belong to a given class). Weber *et al.* [21] number amongst the earliest of authors proposing a solution in this category; they learn to recognise letters of the alphabet. Importantly, they showed it is possible to identify object classes without prior explicit segmentation. Leibe and Schiele [2] continued in this vein by demonstrating it is possible to simultaneously segment and recognise a wider range of object classes. Using a Harris detector [13] to locate the centre of interest regions, they cluster features into a codebook. The spatial relation between these parts is determined relative to a centroid determined by a voting system. Fergus *et al.* [7] adopted a similar approach, but included a greater array of invariants and so produced a system that might be expected to be more robust to changes in scale, orientation and so on (but we are not aware of any authoritative comparison). Similar work continues to date, see [19, 11, 18, 1] for example.

All of this work is premised on photometric features. Yet it is un-controversial to label an object in a drawing or a painting in just the same way one might label a photograph: “Ce n’est pas un pipe” (thanks to Magritte) applies equally to images of all kinds. Accordingly, we would like our visual object classifier to work with images of all kinds. In this paper we propose that image structure should play an important role when classifying objects across image kinds. It is true that graphical structure is included as in many of the models already cited, and is dominant when articulated objects are of interest [6]. Importantly, we can cluster objects of the same class in feature space, regardless image class.

The importance of structure over appearance is evident when we consider our ability to see objects where none exist: in clouds, stains on walls are all too readily interpreted as faces, cars, trains and so on. In all of these cases are difficult to explain without appeal to structure, because the appearance of parts actually observed look nothing like the parts imagined. Interestingly, because the structure of clouds is highly variable it can be argued that seeing clouds exactly as clouds provides a counter example to the importance of structure. We conclude that neither appearance nor structure alone are sufficient to explain the phenomenon of object classification in the round.

The use of structure is not the only novelty in this paper. Our approach is not premised a codebook learned by classifying interest features, as is the greater majority of literature to date, but on watershed regions. Specifically we build a hierarchical description of an image by grouping watershed regions. We chose to group watershed regions because we observed that they tend to isolate features of interest and which are thought to be important for image description: dots, T-junctions, line ends, corners, contrast edges and so on. Some watershed regions enclose no features at all. Consequently watershed regions have the potential to be a useful alphabet that includes whitespace. The fact that watershed regions can vary in shape has proven of little consequence, at least we have not noticed any significant effect.

Watershed regions over-segment an image; we are not the first to group them, but there is necessary novelty in our grouping approach. The arises in part because we seek to acquire objects classes in a semi-unsupervised environment, whereas watershed regions are usually grouped with a view to image segmentation. For example, Haris *et al.* [15] use the watershed of the magnitude of the image derivative. They adopt a hierarchical agglomerative clustering approach which terminates when the cost of grouping any pair rises above a threshold (which varies from image to image). Gauch [10] also segments by clustering watersheds of the derivative image, filtering the image in scale space to do so. Malpica *et al.* [17] build on the work of Haris *et al.* [15] by introducing multidimensional description for regions, rather than the scalar gray level. We also use a multi-dimensional description, but handle it differently from Malpica *et al.* [17], as we will make clear below.

We make two main contributions in this paper:

- We extend watershed clustering to allow multidimensional vector descriptions of regions, and in particular demonstrate the value of decorrelating the vector elements. We also provide a halting criterion that requires no threshold, but is independent of the picture.

- We show that clustering in pattern space places photographs and artwork of the same object into the same category, which provides ground for objection identification.

The first is discussed together in Section 2, the second in Section 3.

2 Grouping watershed regions

This section describes our approach to grouping watershed regions. The aim is to produce a rough segmentation of visual objects. We use hierarchical agglomerative clustering to group; that is we group pairs of regions, thereby creating a new region. The grouping method is described in Subsection 2.1. We could continue until just one region remains but have a principled way to halt the grouper; Subsection 2.2. Having described the structure of our procedure we then demonstrate the value of using a decorrelated feature vector, in which we acquire a decorrelation matrix via supervised learning, see Subsection 2.3 and Subsection 2.4.

2.1 Grouping Method

We use Luc and Soille's watershed algorithm [16] to obtain an initial set of watershed regions. Unlike work aimed to segment images, see [15, 10] for example our watersheds are computed directly from the intensity image, not from it derivative magnitude. Our grouping method is based on the work of Haris *et al.* [15]. They define the cost of grouping a pair of regions as $N_i N_j/(N_i+N_j)(\mu_i-\mu_j)^2$ in which μ is the average gray value over the region of N pixels. They group adjacent regions with the smallest error value, creating a new region from the union of the regions and their boundary. This continues until the cost exceeds a pre-computed threshold which is based on the noise distribution of the image. We note that this threshold is image dependent and that noise estimates are difficult to compute and often unreliable.

As mentioned, Malpica *et al.* [17] have used multi-variate descriptions of watershed regions based on the same grouper. Their extension uses the mean of a set of feature vectors in a region, μ. It redefines the cost as proportional to the sum the root of absolute differences of individual elements between means: $\sum_k |\mu_{ik}-\mu_{jk}|^{1/2}$, the scale factor $N_i N_j/(N_i+N_j)$ is the same. We take a very different stance: we consider the distribution of features vectors across a region, then enquire into the dissimilarity between two distributions.

We begin by supposing that each pixel $\mathbf{x}$ in a region, $\mathscr{R}$, supports a vector of measures $\mathbf{v}(\mathbf{x})$; the vector is typically formed by filtering with a filter bank. Next we approximate the distribution of the vectors in a region with a Gaussian. For the *i*th region we compute the number of pixels, the mean, and the covariance:

$$N_i = |\{\mathbf{x} \in \mathscr{R}_i\}| \tag{1}$$

$$\mu_i = \frac{1}{N_i}\sum_{\mathbf{x}\in\mathscr{R}_i} \mathbf{v}(\mathbf{x}) \tag{2}$$

$$\mathbf{C}_i = \frac{1}{N_i}\sum_{\mathbf{x}\in\mathscr{R}_i} (\mathbf{v}(\mathbf{x})-\mu_i)(\mathbf{v}(\mathbf{x})-\mu_i)^T \tag{3}$$

Together these terms make up an eigenmodel. We define the approximate error for each watershed region $\mathscr{R}_i$ which is $\mathbf{H}(\mathscr{R}_i) = \sum_{x\in\mathscr{R}_i} (\mathbf{v}(\mathbf{x})-\mu_i)^T(\mathbf{v}(\mathbf{x})-\mu_i)$. The ap-

proximation error for the whole picture is, can defined as $\mathbf{W}(\mathbf{P}) = \sum_{\mathscr{R}_i \in \mathbf{P}} \mathbf{H}(\mathscr{R}_i)$ where $\mathbf{P} = \mathscr{R}_1, \mathscr{R}_2, ..., \mathscr{R}_k$ is a k partition for the image.

Finding the optimal partition $\mathbf{W}_{min}$ can be transformed into a stepwise merging process(a greedy algorithm) which merges two watershed regions at each time which has minimum value of cost for merging them. The cost of grouped a pair of regions is

$$e_{ij} = \frac{N_i N_j}{N_i + N_j} (\mu_i - \mu_j)^T (\mu_i - \mu_j) \tag{4}$$

when two regions are merged we can efficiently compute a new Gaussian:

$$\mathbf{C}_{\text{new}} = \frac{N_i}{N_i + N_j} \mathbf{C}_i + \frac{N_j}{N_i + N_j} \mathbf{C}_j + \frac{N_i N_j}{(N_i + N_j)^2} (\mu_i - \mu_j)(\mu_i - \mu_j)^T \tag{5}$$

there is need for us to return to the image data. The final term allows for the difference between means. It can be thought of as measuring the volume of the symmetric difference of two classes. Its trace is proportional to the error as we define it, and is also related to the errors defined by Haris *et al.* [15] and Malpica *et al.* [17].

As the regions merge the feasibility of the Gaussian assumption declines. This is acceptable because so we are willing to accept broader approximations of larger region, which is in line with scale-space descriptions. Alternatively, the size of the approximation is appropriate to the size of the region.

2.2 Stopping Criterion

Now we turn our attention to the halting criterion. Rather than use a threshold we appeal to spectral graph theory [3] to halt the merging process. Each step of the grouping process produces a new partition of the original watershed regions. The elements in the partition delimit regions of the image which, we hope, correspond to visual objects. At least, we desire the partition to *roughly* segment visual objects — recall that our aim in object class identification, not image segmentation.

The watershed regions in any element in a partition form a graph of nodes and arcs: watershed regions are nodes, arcs connect adjacent regions. Hence each partition has an associated structure inherited from the watershed transform. We make use if the structure within partition elements to develop a halting criterion; we assume there is no connectivity between sections of the partition (because the image comprises the union of independent objects).

Our halting criterion uses the Laplacian energy of a graph [12]. For a graph $G = (V,E)$, where V is the node sets and E is the arc relationship among the nodes V. We can construct the adjacency matrix A for the graph, the Laplacian matrix L is defined as $L = D - A$, where D is the degree matrix. The Laplacian energy for graph G is defined as

$$E(G) = \sum_{i=1}^{|V|} \left| \lambda_i - 2\frac{m}{|V|} \right| \tag{6}$$

In which: the λ_i are eigenvalues of the Laplacian matrix L; m is the sum of the arc weights over the whole graph, which is just the sum of the elements in matrix A divided by 2; $|V|$ is the number of nodes in graph. The arc weights are defined here as a function

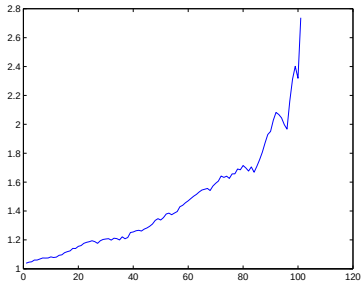


Figure 1: An example of a partition energy plot as a function of iteration number. We halt the grouping process at the right-most local minimum.

of the the merging error in Equation 4 $w_{ij} = \exp(-e_{ij}/e_{\max})$. The normalising factor, $e_{\max}$, is the maximum error over the all the original watershed regions, not just those in a particular element (region, a putative object) of a particular partition (regions collected into an image). It is therefore a constant for a given picture.

The $E(G)$ gives the energy of an element in a partition. We define the energy of the partition to be the average *normalised* Laplacian energy over all N elements of the partition:

$$\xi = \frac{1}{N}\sum_{i=1}^{N}\frac{E(G_i)}{|V_i|} \tag{7}$$

We use normalised Laplacian energy $E(G)/|V|$ to estimate the average connection energy per node; this is rather like normalising spatial measures (such as image moments) by area.

Now, the partition energy ξ is a function of iteration number — simply because each step of the grouping process yields a new partition; therefore we should write $\xi(k)$. Halting conditions occur when $\xi(k)$ is a local minimum, that is $d\xi(k)/dk = 0$ and $d^2\xi(k)/dk^2 > 0$. Figure 1 shows that there is more than one such local minimum. We currently select the last of those, this gives a partition with the fewest number of regions.

2.3 Decorrelating feature vectors

This grouping method as described so works reasonably well, but performance is improved by decorrelating terms in the vector $\mathbf{x}$. We use a decorrleating linear transform, a square matrix $\mathbf{K}$, so that

$$\mathbf{x} \mapsto \mathbf{K}\mathbf{x} \tag{8}$$

Now the error is proportional to $(\mu_i - \mu_j)^T\mathbf{K}^T\mathbf{K}(\mu_i - \mu_j)$. Now, since $\mathbf{x}^T\mathbf{A}\mathbf{x}$ can be written as an inner product $\mathbf{A}_{ij}(\mathbf{x}\mathbf{x}^T)_{ij}$ (using tensor notation) we see that the error is just a linear combination of the all terms in the difference between the means, not just the diagonal.

The rational underpinning decorrelation arises from the desire for a small inner product in the cost term of Equation 4. Clearly, orthogonal vectors give a zero inner product, therefore it is sensible used decorrelate feature vectors when compute the cost.

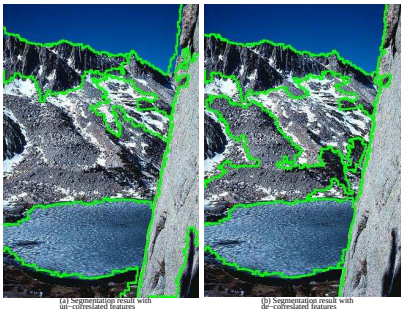


Figure 2: Left Image: Segmentation result with uncorrelated features; Right Image: Segmentation result with decorrelated features.

The decorrelation matrix **K** comes via supervised learning: users pick pairs of adjacent regions they wish to be grouped in 10 different pictures. During training, a user may group in hierarchical fashion. We recorded all pairings (about 1500 in total) and thereby obtain a training collection of $\mathbf{u}_{ij} = (\mu_i - \mu_j)$ vectors. Supervised training helps ensure that decorrelation has at least some grounding in human perception – and using a trained **K** does improve grouping — but we are clear in making no stronger claim about **K** than this.

The feature de-correlation matrix **K** is computed using a fixed-point based variant of independent component analysis (ICA) due to Hyvärinen [14]; which is computationally very efficient yet statistically robust. ICA differs from PCA in that, ICA yields statistically independent components rather than simple decorrelation. Given the set of input training vectors $\mathbf{u}_{ij} \in \Re^n$, we determine a $n \times n$ matrix **K**, such that the components of the mapped vector **Kx** are mutually independent.

2.4 Grouping Results

In all of the results shown, a 4-dimensional feature vector $f = (r,g,b,v)$ for each pixel was used, where r,g,b are just the colour channels of the image and v being the gradient magnitude. Results of our grouper are shown in Figure 2 and 3. Within each Figure, we compare differences in the final segmentations using uncorrelated and decorrelated features.

In Figure 2, the left image corresponds to the final segmentation result using uncorrelated features, while the right image used decorrelated features. It is obvious that result on the right segments out the mountain on the right and the lake well, plus a clearer separation of the mountain at the back. Again, in Figure 3, the segmentation on the right (using decorrelated features) has a better segmentation on the house, car, grass and mountain.

3 Clustering in Pattern Space

In this section, we explain how objects classes can be automatically learned from the hierarchical descriptions output by our grouper. The significant novelty is the use of structure as a feature in its own right, which means we form object classes using images of different kinds: photographs, paintings, and drawings.

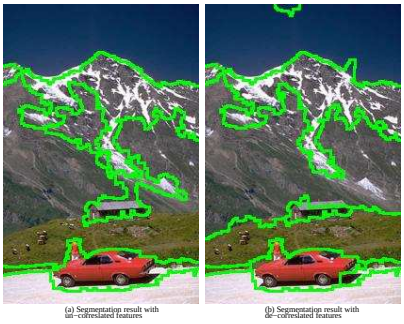


Figure 3: Left Image: Segmentation result with uncorrelated features; Right Image: Segmentation result with decorrelated features.

The idea here is the structure of an object — the structure output by the grouping process — is more or less invariant to changes of image type: the structure of a face in a photograph is the same as the structure of a face in a drawing. We make use of methods from graph spectral theory that encode structures, even noisy structures, into a pattern space of fixed dimension. This furnishes us with the opportunity to cluster objects on the basis of their structure. As the experiment below shows, we are able to cluster objects in a single class regardless of the kind of image they come from.

Given a root node output by the grouper (that is, an element in the final partition) we form a feature vector for it. This feature vector encodes information regarding both content and structure. Content information is held in the eigenmodels associated with every region in the merging process. Recall that this describes decorrelated feature distributions rather than photometric data. The eigenvalues of the covariance matrix for the node are taken to represent its content. Structural information comes from the hierarchy generated by the merging process. Each root note expands into a binary tree; we compute the eigenvalues of the Laplacian matrix of this tree. Note that this Laplacian differs from that used for the halting condition: this one uses a tree output by a process rather than a spatially-dependent graph.

We form a vector of fixed size by choosing a fixed number of dominant eigenvalues both content and structure; we use the largest four from each. Now we have a vector of fixed size to describe root nodes. The structural component of this compound vector is particularly interesting: it is robust in the sense that graphs extraneous or ”noisy” arcs and nodes all project to about the same point. This is a very useful property because it means the result of this projection is robust to grouping error and also to watershed segmentations. Finally, the feature vector for each root node is subject to a multidimensional scaling transform [5] to project it into three dimensions. This reduction of dimension mitigates the ”curse of dimensionality” problem given sparse data, as we have. We use a Gaussian Mixture Model [9] to cluster feature vectors for root nodes, which is a fully automatic method — including choosing the number of clusters.

We now describe our object class identification experiments. We selected two different objects, faces and leaves. For each object we collected real world photos from the Oxford-Caltech database [8], and also paintings and drawings. Figure 4 shows the examples images. The grouper fully automatically partitioned each image. Then we extracted the feature vectors for each root node of the partition, and clustered them. Figure 6 shows

Figure 4: Original faces and leaves (Pictures, Painting & Drawing)

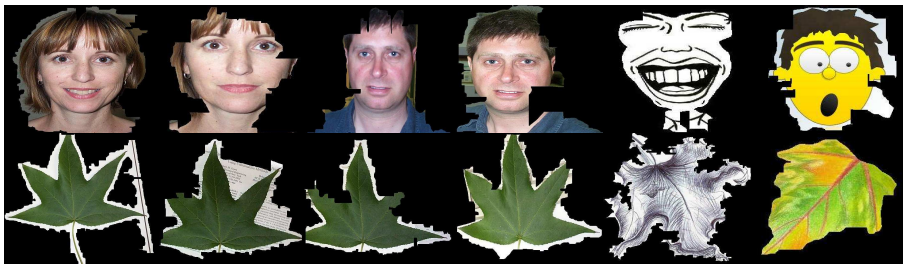


Figure 5: Segmented faces and leaves

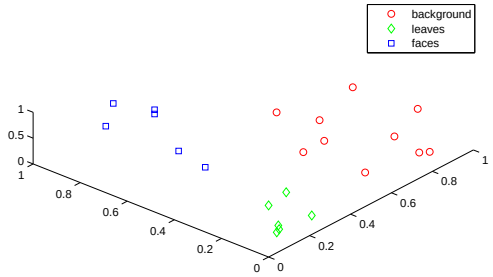


Figure 6: faces and leaves recognition results by using the Gaussian mixture model

the results; the leaves, faces and general background are well separated. To separate foreground from background, a user named(by pointing) each class; the nodes in the leaf and face class were then rendered to give Figure 5. As can be seen the faces and leaves have not been segmented from the background correctly but also clustered regardless of their image class.

4 Conclusions

This paper introduces a novel way to group watershed regions; it demonstrates the value of decorrelating feature vectors by training, and provides a robust halting condition that is picture independent. The output of the grouper is a set of regions, each of which expands

into a structure. We have encoded that structure in a vector of fixed length. This allowed us to cluster objects of the same class but from different image types. Except for the training stage of the grouper, the fact we use a semi-unsupervised approach for learning object class clusters, and the need to identify background explicitly our method is fully automatic. Wide variance in background is expected, whereas objects are expected to show much lower variance. This may be exploited in future work to remove the need for an user to differentiate between them.

The use of structure clearly enables objects from different image classes to be classified as one. It is not at all clear that alternative approaches to learning object classes can do likewise, because they are photometrically based. It is true that some use curves as the basis of class identification [1], but these curves are fitted to contrast edges in a photograph — this is not a suitable approach for line drawings. Watershed regions, on the other hand, tend to isolate regions of interest is all classes of image; as we observed at the top of this paper watershed regions provide a useful alphabet that include whitespace (so whitespace appears in our binary tree).

We believe this paper supports a considerable body of future work, an empirical comparison with contemporary methods for object class identification being just one direction. The ability to classify objects across image classes has implications for content based retrieval as well. In the future, we aim to improve the grouping process by investigating other means of feature vectors, which we hope will lead to better segmentations. Furthermore, we will test on more objects to evaluate the robustness of our algorihm.

5 Acknowledgements

This work was supported by EPSRC grant EP/D05429X/1

References

[1] A. Pinz A. Opelt and A. Zisserman. A boundary-fragment-model for object detection. *ECCV*, 2:575–588, 2006.

[2] B. Leibe asnd B. Schiele. Interleaved object categorization and segmentation. In *BMVC*, volume 2, pages 759–768, 2003.

[3] F. R. K. Chung. *Spectral graph theory*. American Mathematical Society, 1997.

[4] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. In *European Conference on Computer Vision*, volume 1407 of *Lecture Notes in Computer Science*, pages 484–, Freiburg, Germany, 1998. Springer-Verlag.

[5] T. Cox and M. Cox. *Multidimensional Scaling*. Chapman and Hall, 1994.

[6] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 1(61):55–79, 2005.

[7] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, 2003.

[8] R. Fergus, P. Perona, and A. Zisserman. Objects class recognition by unsupervised scale-invariant learning. *IEEE Conference on Computer Vision and Pattern Recognition*, 2:264–271, 2003.

[9] M. A. F. Figueiredo and A. K. Jain. Unsupervised learning of finite mixture models. *IEEE PAMI*, 24(3):381–396, 2002.

[10] J.M. Gauch. Image segmentation and analysis via multiscale gradient watersheds. *IEEE Transactions on Image Processing*, 8(1):69–79, 1999.

[11] K. Grauman and T. Darrell. Unsupervised learning of catagories from sets of partially matching image features. In *CVPR*, 2006.

[12] I. Gutman and B. Zhou. Laplacian energy of a graph. *Linear Algebra and its Applications*, 44:29–37, 2006.

[13] C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. 4th Alvey Vision Conference*, pages 189–192, Manchester, UK, 1998.

[14] A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.

[15] K.Haris, S.Efstratiadis, N.Maglaveras, and A.Katsaggelos. Hybrid image segmentation using watersheds and fast region merging. *IEEE Trans. Image Processing*, 7(12):1684–1699, 1998.

[16] V. Luc and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, 1991.

[17] N. Malpica, J.E. Ortuño, and A. Santos. A multi-channel watershed-based algorithm for supervised texture segmentation. *Pattern Recognition Letters*, 24:1545–1554, 2003.

[18] K. Mikolajczyk, B. Leibe, and B. Schiele. Multiple object class detection with a generative model. In *CVPR*, 2006.

[19] J. Mutch and D. Lowe. Multiclass object recognition with sparse, localized features. In *CVPR*, 2006.

[20] M. Alex O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *ECCV*, pages 447–460, 2002.

[21] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for visual object class recognition. In *ECCV, volume 1842 of Lecture Notes in Computer Science*, pages I:18–32, 2000.

Simple Representation and Approximate Search of Feature Vectors for Large-Scale Object Recognition

Koichi Kise, Kazuto Noguchi, and Masakazu Iwamura
Osaka Prefecture University, Japan
{kise,masa}@cs.osakafu-u.ac.jp, noguchi@m.cs.osakafu-u.ac.jp

Abstract

This paper presents two methods of large-scale recognition of planar objects with a simple representation and approximate search of local feature vectors. A central problem of the use of local feature vectors is the burden of computation and memory for finding nearest neighbors. To solve this problem, the proposed methods embody the following: (1) a simple bit representation of feature vectors and hashing enable us to fast access with less memory, (2) approximate search with query perturbation allows us to find approximate nearest neighbors efficiently. From large-scale experiments using 10,000 objects in the database and 2,000 query images, it was found that only 10–20% of correct nearest neighbors were enough for achieving recognition rate of 98.0%. The processing time for achieving this rate was 8.3 ms / query (excluding time for feature extraction). We have also tested the scalability of a proposed method using the database of 100,000 objects and obtained the result of 92.3% accuracy in 4.5 ms /query.

1 Introduction

Local feature vectors with high stability and discriminability such as SIFT [8] have been opening a new vista of robust and accurate object recognition. Extensive efforts have so far been made on, for example, recognition of planar objects [6, 9], 3D objects and their parts [4], object and scene categories [7], and structuring videos [10]. This paper is concerned with the application of PCA-SIFT [5] to the task of large-scale recognition of planar objects such as posters, book covers, and pictures of natural scenes, people, artifacts, etc. The objective of such recognition is to use the recognition technology for *tagging* planar objects in the real world: the user can deal with planar objects like *barcodes* for accessing their relevant information. Moreover the user is capable of establishing his/her own *links* that are clickable using mobile cameras.

One of the most prominent ways of the use of local feature vectors is often called *bag-of-words* or *bag-of-features* which represent images as a set of *visual words* obtained by clustering feature vectors [10]. When we scale up the number of objects to be recognized by the bag-of-features approach, at least the following problems need to be addressed. The first one is how to improve the discriminability of visual words. It is reported in the literature (e.g., [9]) that better performance is obtained by a larger number of visual

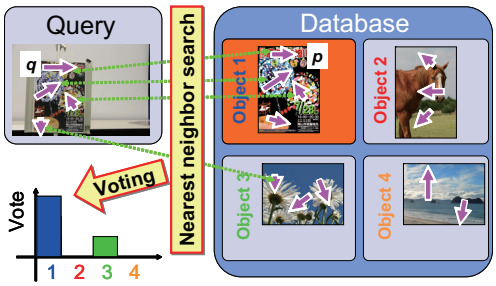


Figure 1: Object recognition by voting.

words, which increase the computational burden for indexing, storage and retrieval. Some researchers have attempted to solve this problem by applying a tree structure designed for dealing with local feature vectors [9]. The second problem is recomputation of visual words; if a number of new objects are added to the database, it would be necessary to recompute all visual words from scratch for better indexing. The last problem is about minority feature vectors. If some objects are mainly described by such minorities, it is difficult to recognize them correctly because few visual words appropriately represent such objects.

A simple way to solve the second and the third problems is just to take all feature vectors as visual words for indexing, which is an approach called memory-based or case-based vision. The input (query) image is recognized by matching its feature vectors to those in the database. In this scenario, *nearest neighbor search* of feature vectors is a fundamental processing. However this exacerbates the first problem. For the recognition of 100,000 objects, for example, the number of feature vectors is more than a hundred million and the number of times of matching with a simple sequential search exceeds 100 billion for recognizing a single query image; it is prohibitive to keep the recognition efficient. Thus the important problem here is how to deal with a huge number of feature vectors for storage and retrieval.

The main contribution of this paper is to solve this problem by (1) a simple bit vector representation of feature vectors and their hashing for efficient storage and retrieval, (2) an *approximate* nearest neighbor search with query perturbation for keeping the accuracy of nearest neighbor search enough for object recognition. For example, a proposed method is capable of recognizing planar objects using the database of 10,000 objects with the accuracy of 98.0% in 8.3 ms/query (excluding time for feature extraction). A different setting allows us 92.3% accuracy in 4.5 ms/query for the database of 100,000 objects.

2 Object Recognition by Voting

In this paper, we employ a simple scheme of object recognition as shown in Fig. 1. The task here is to find the corresponding image of an object from the database (DB) by matching the input image captured by a camera to each image in the DB. In this simple model of object recognition, both the camera-captured image, or *query image* and images in the database are described by a number of feature vectors produced by the local descriptor PCA-SIFT.

An outline of the recognition procedure is as follows. For each feature vector $\boldsymbol{q}$ of the query image, the corresponding feature vector $\boldsymbol{p}$ is found in the DB as the approximate nearest neighbor of $\boldsymbol{q}$. As a result, an object to which the vector $\boldsymbol{p}$ belongs obtains a vote from $\boldsymbol{q}$. Finally, the query image is recognized as the object with the largest number of votes.

It is easy to realize that, for such an object recognition scheme, the key for the memory efficiency is how to index and store feature vectors. Because the recognition process is dominated by finding nearest neighbors, nearest neighbor search is the key component to improve overall recognition efficiency. It is desirable to minimize both time and memory required to find nearest neighbors without losing the recognition accuracy.

3 Approximate Nearest Neighbor Search

An effective way of improving the efficiency of nearest neighbor search is to introduce "approximation" to the notion of "nearest neighbors". In this section we briefly describe two major methods: ANN (Approximate Nearest Neighbor) by Arya et al. [2], and LSH (Locality Sensitive Hashing) by Indyk et al. [3], both of which were utilized for comparison in Sect. 5.

ANN is a method of nearest neighbor search that employs a tree structure: each non-leaf node corresponds to a test that splits feature vectors into two disjoint sets, and each leaf node represents a feature vector. A node also corresponds to a cell, i.e., a region in the feature space. ANN traverses the tree structure to collect feature vectors for computing the distance to the query feature vector. The accuracy and the efficiency of computing nearest neighbors depend on the number of feature vectors for distance calculation. The approximation factor ϵ, which defines $(1+\epsilon)$-approximate nearest neighbors and changes the radius of search in the feature space, is to control the number. The smaller number of feature vectors is obtained with the larger ϵ.

LSH is a method of approximate nearest neighbor search using hash tables. We briefly describe an implementation of LSH called E^{2}LSH (Exact Euclidean LSH; simply called LSH for simplicity in this paper) [1]. To store a d-dimensional feature vector, LSH converts it into L different k dimensional integer vectors each of which is recorded in a distinct hash table. To find an approximate nearest neighbor of the query feature vector it is likewise converted into L integer vectors, and then they are utilized to find entries (feature vectors) in L hash tables. The distances from the query feature vector to each retrieved feature vector is calculated and the one with the minimum distance is returned as the result. With the same k, a larger L increases the probability of finding the exact nearest neighbor but also increases the computation time as well as the required memory.

4 Proposed Methods

4.1 Concepts

Both ANN and LSH are methods for finding the approximate nearest neighbor of a single vector. The simplest way of using these methods for object recognition is to apply them for each query feature vector to find its correspondence for vote. It is worth noting that our objective is not to find the best candidate for each feature vector, but to recognize

objects by voting. Thus as long as objects are correctly recognized, there should be no problem even with a certain amount of votes for wrong objects. This means that there is still some room of drastic approximation for improving the efficiency of recognition.

The most time-consuming part of object recognition by voting is the distance calculation with a large number of feature vectors. Thus the efficiency of recognition is improved by reducing the number. The most part of memory required for the recognition is occupied for recording a large number of feature vectors. Thus the memory requirement is eased by reducing the amount of memory for recording feature vectors.

In this paper we attempt to apply some simple methods for this purpose. A major drawback of the use of LSH for a large-scale object recognition is the memory consumption by multiple hash tables. The proposed methods employ only a single hash table to solve this problem. To compensate the lack of multiple hash tables, we introduce perturbation of query feature vectors that simulates errors caused by different imaging conditions. With this common strategy, we propose two methods for object recognition: the method with distance calculation (Method A) first retrieves feature vectors close to the query feature vector and then find the nearest one for voting by the distance calculation. The method without distance calculation (Method B) is just to skip the computation of the nearest feature vector of the method A; it simply votes for all feature vectors with a hope that the correct object accumulates the maximum votes. The details are explained in the following.

4.2 Indexing of Feature Vectors

In order to index feature vectors in the hash table, it is required to convert real-valued feature vectors into integers. A common solution is to apply the vector quantization. However, in order to keep our methods simple and efficient, we apply scalar quantization as follows. Let $\boldsymbol{p}$ be a n-dimensional feature vector $\boldsymbol{p} = (p_1, p_2, ..., p_n)$ obtained by applying PCA-SIFT. We binarize each dimension by

$$u_j = \begin{cases} 1 & \text{if } p_j \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

to produce a bit vector $\boldsymbol{u} = (u_1, ..., u_d)$ where the first $d(\leq n)$ elements are employed for the indexing. The threshold 0 is reasonable for values obtained by PCA-SIFT, because the their averages are around 0. Then the following hash function

$$H_{\text{index}} = \left(\sum_{i=1}^{d} u_i 2^{(i-1)} \right) \mod H_{\text{size}} \tag{1}$$

is applied to obtain the hash value of $\boldsymbol{p}$ and store it at the corresponding slot of the hash table.

For the method A with distance calculation it is necessary to record the original feature vectors. For the method B without distance calculation, on the other hand, it is unnecessary to keep original vectors. This greatly helps reducing the amount of memory.

Multiple vectors with the same hash value are stored by chaining. If the length of the chain exceeds the threshold c, all entries with the same hash value are deleted. This is the strategy called “stopword elimination”. Although it is simple, it improves the memory consumption as well as the accuracy of recognition.

4.3 Method with Distance Calculation (Method A)

Next we describe the object recognition by using the method A. The elementary task is to retrieve a set of feature vectors $\{\boldsymbol{p}\}$ close to the query vector $\boldsymbol{q}$ from the hash table. Then the method finds the nearest neighbor $\boldsymbol{p}^*$ of $\boldsymbol{q}$ in $\{\boldsymbol{p}\}$ and votes for the object to which $\boldsymbol{p}^*$ belongs. The object in a query image is recognized by voting with all feature vectors extracted from it.

The most important step here is to find the set $\{\boldsymbol{p}\}$, which should include the nearest feature vector. A way is to access the hash table based on the hash value obtained from the query vector. In this case feature vectors having the same hash value are retrieved. Unfortunately it is unsatisfactory since variation of a query vector may change its bit vector, which results in accessing the slot of the hash table that does not contain its nearest neighbor.

In order to ease this problem, the query vector is expanded into several bit vectors to find its nearest neighbor. Because the query vector is transformed into the bit vector using the threshold of 0 for each dimension, the query vector having values close to 0 at some dimensions can be converted into the bit vector different from its nearest neighbor. We simply utilize the error range e as a parameter for generating different bit vectors. For the dimension q_j satisfying $|q_j| \leq e$ in the query vector $\boldsymbol{q} = (q_1, ..., q_d)$, the other bit value $u'_j = (u_j + 1) \bmod 2$ (if $u_j = 0$ then 1, otherwise 0) is also utilized to generate bit vectors. This means that the error of the value is estimated within the range defined by e. For the query vector $\boldsymbol{q} = (-10, 100, 2)$ and $e = 5$, for example, two bit vectors (0,1,1) and (0,1,0) is employed for the access of the hash table.

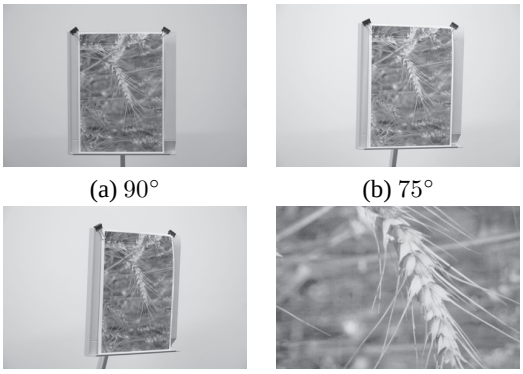
Unlimited application of this strategy increases the expanded bit vectors exponentially. In order to avoid this problem we utilize the limit b of the number of dimensions for the application. The method applies the strategy for q_j from $j = 1$, i.e., the dimension with the largest eigenvalue. If the number of dimensions that satisfy the threshold e exceeds the limit b, the application stops and the remaining dimensions are simply utilized as they are. In other words, the number of expanded bit vectors for a query vector is at most 2^b. For the above example of $\boldsymbol{q}$ with $e = 20$ and $b = 1$, two bit vectors (0,1,1) and (1,1,1) are obtained.

4.4 Method without Distance Calculation (Method B)

The method B shares most of the processing with the method A. Thus the parameters of the processing are also c, e, b and d. The difference is as follows. The method A computes the distance from the vector $\boldsymbol{q}$ to each element of the set $\{\boldsymbol{p}\}$ to find the nearest neighbor of $\boldsymbol{q}$ in the set. The method B, on the other hand, skips this calculation and simply votes for all vectors in the set $\{\boldsymbol{p}\}$. Because there is no need to calculate the distance, it is not necessary to record the original feature vectors. This enables us to reduce drastically the amount of memory needed for the processing.

(a) (b) (c)

Figure 2: Examples of images in the database



(a) 90° (b) 75°

(c) 60° (d) Part

Figure 3: Examples of query images. (a) – (c) covers the whole area, and (d) covers only 1/4 of the whole area. (a) and (d) are frontal view (90 °), while (b) and (c) are cross shots.

5 Experiments

5.1 Experimental Settings

The objective of experiments is to clarify the following relations: (1) relation between the accuracy of nearest neighbor search and that of object recognition, (2) relation between the accuracy of object recognition and the processing time, (3) relation between the number of objects to be recognized and the amount of memory.

As the images in the database, we have prepared in total 100,000 images collected using Google image search (Fig. 2(a)) and Flickr (Fig. 2(b)), as well as images available at the site of PCA-SIFT [1] such as Fig. 2(c). Images were resized to have their longest side less than 640 pixels. The average number of feature vectors extracted from an image was about 2,000.

Images used as queries were prepared as follows. We first selected at random 500 images from the database. Then these images were printed out onto A4 paper and converted into images by taking their pictures in four different ways as shown in Fig. 3. The size of the images were reduced to 512×341 pixels and then PCA-SIFT was applied to extract feature vectors. About 600 feature vectors were obtained on an average from an image.

[1] http://www.cs.cmu.edu/~yke/pcasift/

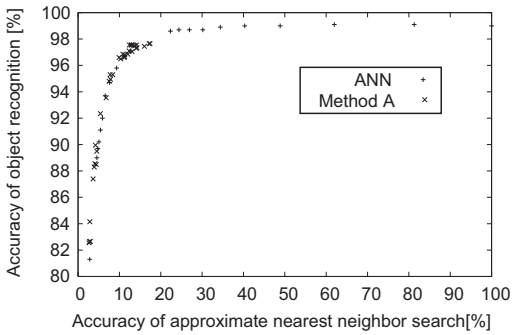


Figure 4: Relation between the accuracy of nearest neighbor search and that of object recognition.

The methods A and B were compared with methods of object recognition using ANN and LSH as engines of approximate nearest neighbor search [2]. Parameters for each method were set as follows. The number of dimensions of feature vectors for ANN and LSH was set to 36. For ANN, the approximation factor ϵ was changed from 2 to 1000. Although it may seem extremely large, we show that it is meaningful at least up to $\epsilon = 10$. For other parameters, the default values were employed. It means, for example, that we employed standard search with kd-trees. LSH has two major parameters, the number of hash tables L and the number of dimensions k for indexing feature vectors. We tested it by combining the values $L = 1, 3, 6, 15, 28$ and $k = 20, 24$. For the methods A and B, we have four parameters, i.e., the number of dimensions m for indexing feature vectors, the number of collisions c for deleting feature vectors, the error range e of a feature value, and the limit b of the number of dimensions for the expansion of the original bit vector. The tested ranges were as follows. $m = 24, 26, 28$, $c = 1, ..., 100$ and ∞, $e = 200, 500, 1000$, and $b = 0, ..., 15$.

In the following results, computation time excludes the time required for extracting feature vectors. We employed a computer with AMD Opteron 2.8GHz CPU and 16GB RAM.

5.2 Accuracy of Approximate Nearest Neighbor Search

For measuring the relation between the accuracy of nearest neighbor search and that of object recognition, we applied methods A and that with ANN using the database including 10,000 images. The accuracy of nearest neighbor search was measured as the ratio of the number of *exact* nearest neighbors to the total number of feature vectors.

Figure 4 shows the results where points were obtained by changing the parameters. We can see the consistency of the results obtained by the two methods. It is also shown that if the accuracy of nearest neighbor search is more than 20%, there is a little impact on the accuracy of object recognition. In addition, only 5% accuracy of nearest neighbor search is needed to achieve the object recognition rate of 90%. This means that we have a large margin of approximation for reducing the computational cost.

[2]The implementations utilized in the experiments were provided by authors of these engines: ANN at `http://www.cs.umd.edu/~mount/ANN/` and LSH at `http://www.mit.edu/~andoni/`

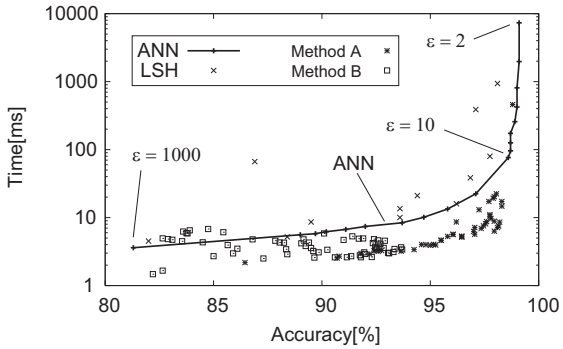


Figure 5: Accuracy of recognition and time for the database of 10,000 images.

5.3 Accuracy of Recognition and Time

Next we evaluated the accuracy of recognition and the time required for achieving the accuracy. Figure 5 illustrates the results. In this graph each point represents a result of a method with a set of parameter values.

The results by ANN are shown with the solid line: the upper right end indicates the result with $\epsilon = 2$ (accuracy:99.1%, time:7.3×10^3 ms) and the other end corresponds to the result with $\epsilon = 1000$ (81.3%, 3.6 ms). Because the accuracy 98.6% in 76.3 ms was obtained with $\epsilon = 10$, ANN was successful to cut down the computation time up to this value. With $\epsilon > 10$, the computation time can be further cut down but it comes with a price of lowering the accuracy.

Next, taking the results by ANN as the baseline, let us discuss results by other methods Most of the results obtained by LSH were inferior to those by ANN. This was mainly due to the cost of using multiple hash tables. The method A, on the other hand, was capable of achieving the processing about three times faster than the method with ANN. Unless the user is interested in results with very high accuracy, the method A is a good choice. For example, 98.0% accuracy was obtained in 8.3 ms/query. If the user is interested mainly in fast recognition while there is no problem with the accuracy around 90%, the method B can be recommended: e.g., 93.1% was in 1.75 ms/query.

5.4 Effect of the Number of Objects

We also tested the effect of the number of objects to the accuracy, the computation time and the memory consumption. The memory consumption was almost proportional to the number of objects in the database. For the recognition with ANN, 16GB RAM ran out for the database of 20,000 objects. The situation is better for the method A because of the use of deletion, and worse for LSH due to the use of multiple hash tables. The best was the method B that utilizes no original feature vectors. The amount of memory for the recognition of 100,000 objects was 7GB.

Time and accuracy using the method B are shown in Fig. 6. As shown in this figure, the time was sub-linear to the number of objects in the database. It is also shown that the accuracy almost stayed unchanged. These results indicate the scalability of the method B.

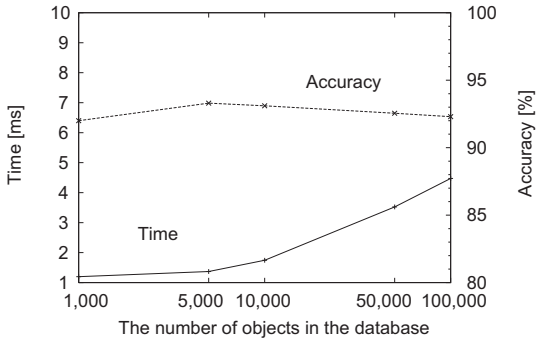


Figure 6: Relation among the number of images in the database, the accuracy of recognition and the processing time, all of which were measured using the method B with the parameters $e = 200$, $d = 28$, $b = 5$ and $c = 5$.

6 Related Work

We focus here on methods for large-scale recognition of planar objects. Ke et al. have proposed a method of sub-image retrieval for near-duplicate detection taking LSH as an engine for approximate nearest neighbor search. The database they employed for experiments included about 20,000 images. Although the size of the database is large, the task was much easier than ours. Their task of retrieval takes as queries automatically generated near-duplicates, while our task is with camera-captured images as queries. For our setting, it is required to cope with much wider variations on images.

Closer tasks are found in the paper by Nister et al. [9], which describes the vocabulary tree for indexing and retrieval of feature vectors. The computation time was reported as 25ms for a database of 50,000 images. They also evaluated their method using 6376 images and reported that the best result was obtained with 10M visual words. This seems almost the size to keep all feature vectors as distinct visual words. In such a case the tree structure is only to find approximate nearest neighbors efficiently. Another database was with 1 million images consisting of all frames from several movies. Although the size was extremely huge, the number of visual words should be much less because most of a sequence of images contains same objects under equal lighting conditions.

As compared to these methods as well as methods with ANN and LSH, the proposed methods can be characterized as follows. A simple representation by the bit vectors and the hashing mechanism improve the efficiency of storage and retrieval. Query perturbation enables us to keep the accuracy of recognition with such simple representation. The number of images used in the experiments and the number of feature vectors employed for the recognition were equivalent or even more than those employed for the above methods.

7 Conclusion

In this paper, we have proposed two methods for the use of individual feature vectors as indices of objects to be recognized. A simple bit vector representation of feature vec-

tors and search methods for their approximate nearest neighbors with query perturbation enable us to keep the processing efficient. From the experimental results using up to 100,000 objects it has been shown that the proposed methods are superior to recognition using ANN and LSH. The future work includes further improvement of accuracy and time based on the combination of nearest neighbor searchers. Recognition of 3D objects is another task to be pursued.

Acknowledgment

This research was supported in part by the Grant-in-Aid for Scientific Research (B) (19300062) from Japan Society for Promotion of Science.

References

[1] Alexandr Andoni, Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab Mirrokni. Locality-sensitive hashing using stable distributions. In Gregory Shakhnarovich, Trevor Darrel, and Piotr Indyk, editors, *Nearest-Neighbor Methods in Learning and Vision*, pages 61–72. The MIT Press, 2005.

[2] Sunil Arya, David M. Mount, Ruth Silverman, and Angela Y. Wu. An optimal algorithm for approximate nearest neighbor searching. *Journal of the ACM*, 45(6):891–923, 1998.

[3] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proc. of the 20th annual symposium on Computational Geometry*, pages 253–262, 2004.

[4] Boris Epshtein and Shimon Ullman. Identifying semantically equivalent object fragments. In *Proc. CVPR2005*, pages 2–9, 2005.

[5] Y. Ke and R. Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *CVPR2004*, volume 2, pages 506–513, 2004.

[6] Y. Ke, R. Sukthankar, and L. Huston. Efficient near-duplicate detection and sub-image retrieval. In *MM2004*, pages 869–876, 2004.

[7] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bag of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. CVPR2006*, pages 2169–2178, 2006.

[8] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[9] David Nistér and Henrik Stewénius. Scalable recognition with a vocabulary tree. In *Proc. CVPR2006*, pages 775–781, 2006.

[10] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proc. ICCV2003*, volume 2, pages 1470–1477, 2003.

Enforcing 3D Constraints To Improve Object and Scene Recognition

Robin Hewitt
Hewitt Consulting, San Diego, CA, US
rhewitt@acm.org

Luis Goncalves
Evolution Robotics Retail, Pasadena, CA, US
luis@evoretail.com

Mario Enrique Munich
Evolution Robotics, Pasadena, CA, US
mario@evolution.com

Abstract

This paper presents an extension to David Lowe's well-known object recognition algorithm based on his Scale and Feature Invariant Transform (SIFT). One of the benefits of Lowe's SIFT-based method is that it can recognize objects from only three keypoints. While this capability can be useful in circumstances where the cost of a false negative is high, it's often the case that false positive are an equal, or greater, concern.

We extend Lowe's algorithm by adding the ability to use 3D constraints during matching. These constraints essentially eliminate false positive matches. We combine our extension with the original algorithm to retain the recognition power of the original method while adding significant robustness against false positives, thereby increasing overall classification power. In addition to improving recognition, our extension returns 3D pose information. Yet, it adds very little computational overhead to Lowe's original algorithm.

1 Introduction

Advances to object recognition during the past decade have given rise to new algorithms. One of the most successful of these is the algorithm developed by David Lowe, based on the Scale Invariant Feature Transform (SIFT) [6]. SIFT (or derivatives of it) has been integrated into a number of commercial products, including Sony's Aibo, Bandai's NetTansor robots, and the visual Simultaneous Localization and Mapping (vSLAM) system [4] by Evolution Robotics.

Lowe's method has many attractive qualities. It recognizes multiple objects in query images based on minimal training input. It's simple to use: no specialized expertise, dataset, nor equipment are required to train new models. Discrimination between multiple

learned objects is handled efficiently. It's completely robust to changes in scale and to in-plane rotations, and it accommodates mild perspective distortions that arise from out-of-plane rotations.

Where SIFT has difficulty, is in handling distortions due to strong 3D perspective effects. This shortcoming limits its usefulness in applications that require robustness to these distortions. Strong 3D effects create several problems for SIFT.

The first is a change to keypoint descriptors. SIFT descriptors model local appearance as a histogram of gradient direction. Strong perspective effects warp image regions, changing the descriptors, thereby decreasing the similarity between descriptors in training and query views. Mikolajczyk and Schmid [8] have introduced an affine invariant descriptor that removes the problem. It has been used successfully in a number of applications, including recent methods for 3D object recognition ([1], [11].

Other perspective difficulties for SIFT arise from parallax effects, however. Lowe uses an affine transformation to approximate mild perspective effects. For objects that are essentially flat or smoothly convex, this is a reasonable approximation. For objects with limblike protrusions, however, parallax effects can be important, and these are not well approximated by an affine warp. Figure 2 illustrates this. As the toy bear rotates about its body's centerline (2b and 2d), head and torso remain nearly stationary within the image, but arms and legs are displaced much farther. Other objects that exhibit a similar parallax effect include airplanes, chairs, and tables.

Parallax is also a significant factor in visual SLAM applications. Figure 4 illustrates this. As the camera moves through the furnished living room, keypoints generated by the chair in the foreground move farther between views than do keypoints generated by the more distant fireplace.

The third problem is that, when out-of-plane rotation is large, an affine warp may be a poor approximation, even when the object is flat or convex. This is also a parallax effect. It's most apparent when a large portion of the object's surface is approximately planar (because self occlusion is then low), and the perspective effect is strong.

In this paper, we present an extension to Lowe's algorithm that accommodates these effects. Our extension consists of two changes to the final step in Lowe's algorithm. The first is extremely simple. Instead of approximating perspective effects by an affine transform, we fit a homography. This small change, by itself, is enough to significantly improve recognition scores on relatively flat objects when out-of-plane rotation is large.

Our second change is more substantial. We apply a 3D constraint based on finding two homographies from different bins in Lowe's Hough transform. Two homographies not only constrain 3D pose information, they overconstrain it. This overconstraint on 3D geometry allows us to apply a very reliable error measure based on self consistency. We show how this self-consistency measure can be used to improve overall recognition performance.

These extensions increase SIFT's discrimination power when parallax effects are present with little additional computation overhead. Further, the same method seamlessly accommodates both planar objects and objects with more complex 3D shape.

Since we retain the machinery of SIFT, the choice to apply our extension is easily toggled on and off. It can be applied whenever false positives are a particular concern, and omitted to enable more lenient recognition. When the extension is enabled, it provides 3D pose information in addition to recognition results.

2 Related Work

Early work on 3D recognition used a visual manifold, with 3D objects or scenes represented as images from several viewpoints [9]. Recently, new methods ([1], [11], and [5]) have begun to emerge that explicitly address the problem of 3D object recognition in the presence of strong perspective effects.

In [1], Ferrari et al. use a combination of local features and region growing to model spatial relationships while providing robustness to heavy occlusion and clutter. However, viewpoint invariance still relies on exhaustive comparisons with every training image. Kushal and Ponce [5] extend Ferrari's work. They build an explicit 3D model of each object from stereo views taken at seven to twelve vantage points in a ring around each object. However, the specialized equipment (stereo rig and turntable) needed for training make this approach unsuitable for some applications.

In [7], Lowe presents an extension to his own SIFT-based recognition method in which features from multiple views of an object are linked. He proposes that an alternative extension would be to solve explicitly for 3D structure during matching. This is the approach we take in this paper. Unfortunately, only qualitative testing was done in [7], making it difficult to compare these two approaches.

Our extension to Lowe's method uses planar homographies. Homographies have been used extensively in object recognition methods ([2], [12]). However, the existing homography-based methods that we're aware of rely on locating planar surfaces in an object. Our approach differs, in that we don't require that homographies correspond to planar surfaces. Consequently, we don't rely on being able to locate planar surfaces, nor even on these being present in the image.

3 Methods

Below, in 3.1, we briefly outline Lowe's SIFT-based recognition method. In 3.2, we describe our extension for handling parallax effects. Last, in 3.3, we describe how we combine our 3D extension with Lowe's algorithm to improve recognition performance.

3.1 A Brief Overview of Lowe's SIFT-Based Recognition Method

Lowe's object recognition method [6] is based on correspondences of salient points at multiple image scales. Each salient point is represented by its SIFT descriptor – a circular histogram of gradient directions.

To compare training and query images, K nearest neighbor matching is used to find putative correspondences between keypoints from all training images and keypoints in the query image. Each keypoint-to-keypoint correspondence casts one vote for a combination of training image, pose, and scale. Votes are accumulated in a Hough transform. The Hough bins that receive the most votes constitute candidate hypotheses for the presence of an object at a particular location, scale, and in-plane rotation.

Last, Lowe validates candidate hypotheses by iteratively applying a least-squares fit for an affine transformation to the keypoints in a Hough bin. Outlier keypoint matches are removed at each iteration. Hypotheses that are reduced to fewer than three keypoint matches are rejected as incorrect.

3.2 Our 3D Extension

3.2.1 From Similarity To Homography

Hough transforms are typically used to detect shapes by accumulating votes from widely separated pixel locations. In Lowe's method, however, the Hough transform plays a different role. Instead of accumulating votes for structures within an image, it accumulates votes for Similarity transforms that map keypoints from one image to another. Lowe's Hough transform quantizes 4D Similarity space (t_x, t_y, s, θ). The votes in each Hough bin represent statistical support for the Similarity transform

$$\mathbf{S}_b = \begin{bmatrix} \bar{s}_b \cos\bar{\theta}_b & -\bar{s}_b \sin\bar{\theta}_b & \bar{t_{x}}_b \\ \bar{s}_b \sin\bar{\theta}_b & \bar{s}_b \cos\bar{\theta}_b & \bar{t_{y}}_b \\ 0 & 0 & 1 \end{bmatrix}, \tag{1}$$

where $(\bar{t_{x}}_b, \bar{t_{y}}_b, \bar{s}_b, \bar{\theta}_b)$ give the translation, scale, and orientation coordinates at the center of bin b. In the last step of his recognition method, Lowe upgrades each $\mathbf{S}_b$ from a Similarity to an Affinity.

Our first extension to Lowe's algorithm is very simple. We upgrade to a Homography, rather than to an Affinity. The Homography upgrade could be done in various ways – with a least-squares fit, with RANSAC, and so on. The method we found worked best is to use gradient descent to convert the Similarity into a Homography by minimizing an error term, $f(e)$. The e here is a vector of reprojection errors for each point pair:

$$e = x_2 - \mathbf{H}x_1, \tag{2}$$

where, x_1 and x_2 are the keypoint locations in the training and test images, respectively, and $\mathbf{H}$ is the homography transform. The function $f(e)$ is a sinusoidal damping function that minimizes the effect of outliers at each gradient-descent iteration. The damping function we use is

$$f(e_i) = sgn(e_i) * ln(1 + |e_i|). \tag{3}$$

At each iteration, we discard outliers and add keypoint matches that were not part of the initial Similarity, but which are inliers to the Homography in both location and scale. This process converges very quickly. Typically, two iterations are enough.

When perspective effects are strong, a homography is often a good approximation for objects with a dominant planar surface, and even for many objects that are not planar. Figure 1 demonstrates the improvement from this simple change.

If a large percentage of the training view's keypoints are matched as inliers, a positive result is returned for the training image. If no valid homography is found, we reject this Hough bin. If, however, a homography is found that has some support, but there are many keypoints in the training image that aren't inliers, we continue to the next step – applying a full 3D constraint. This process is described in the next subsection.

3.2.2 Using Two Homographies To Define 3D Pose

For a 3D fit, we could use RANSAC to search for two additional keypoint matches that are not on the homography, compute the fundamental matrix from these points plus the homography, and evaluate the match based on inlier support. In fact, in our first attempt

to enforce 3D constraints for object recognition, we did exactly this. The difficulty here, however, is that the fundamental matrix by itself is a very weak constraint. It only enforces a point to line mapping. In cluttered query images (such as those in Figure 6), many accidental keypoint matches may conspire to support an incorrect epipolar geometry – or, worse, an incorrect training image. Conversely, the correct solution may have relatively few supporting keypoint matches, especially in the presence of heavy occlusion.

Instead, we make use of the fact that we already have Hough bins, sorted by votes. Using the same method as for $\mathbf{H_1}$, we look at one or more additional bins to see if one of them yields a second homography, $\mathbf{H_2}$. To avoid rediscovering $\mathbf{H_1}$, we mask off an epsilon region around each of its inlier keypoints.

From [3], given two planar homographies, $\mathbf{H}_1$ and $\mathbf{H}_2$, the fundamental matrix, $\mathbf{F}$, can be computed as

$$\mathbf{F} = [\mathbf{H}_i\mathbf{e}]_\times \mathbf{H}_i, \tag{4}$$

where $i = 1$ or 2, and $\mathbf{e}$ is the non-degenerate eigenvector of ${\mathbf{H}_2}^{-1}\mathbf{H}_1$. (Here, and below, $[\mathbf{a}]_\times$ represents the 3×3 skew-symmetric matrix s.t. $[\mathbf{a}]_\times \mathbf{x} = \mathbf{a} \times \mathbf{x}$.)

The significant benefit we gain by computing $\mathbf{F}$ in this way is that we can then compute a direct measure of 3D self-consistency error that's *independent* of inlier support to measure goodness of a match. First, we find solutions to Equation 4 for $i = 1$ and 2:

$$\mathbf{F}_1 = [\mathbf{H}_2\mathbf{e}]_\times \mathbf{H}_2, \text{ and } \mathbf{F}_2 = [\mathbf{H}_1\mathbf{e}]_\times \mathbf{H}_1. \tag{5}$$

We then make use of the constraint that a homography $\mathbf{H}$ is compatible with a fundamental matrix $\mathbf{F}$ if and only if the matrix $\mathbf{H}^T\mathbf{F}$ is skew symmetric [3]. In other words,

$$\mathbf{H}^T\mathbf{F} + \mathbf{F}^T\mathbf{H} = 0. \tag{6}$$

We calculate self-consistency error, e, as

$$e = \max\left(\left|{\mathbf{H}_1}^T\mathbf{F}_1 + {\mathbf{F}_1}^T\mathbf{H}_1\right|, \left|{\mathbf{H}_2}^T\mathbf{F}_2 + {\mathbf{F}_2}^T\mathbf{H}_2\right|\right). \tag{7}$$

3.2.3 Virtual Surfaces

Using Lowe's Hough bins, valid, plane-induced homographies can be identified between views of objects that are not themselves planar. Figure 2 shows keypoint matches for two homographies induced via sets of point matches associated with planes that intersect a toy bear. Each homography was initialized by the keypoint matches in a separate similarity bin. In the same way that a Hough transform for lines finds distributed, but collinear, segments in an image, the Hough transform in Lowe's algorithm helps bring out homographies from "virtual surfaces" that may have distributed, rather than localized, support.

3.3 Using 3D Constraints To Improve Recognition

As test results below show, adding our 3D constraint to Lowe's method essentially eliminates false positive matches. This characteristic makes it a useful complement to the original method, in which false positives can be a problem. The question remains, then, how to smoothly integrate our extension with Lowe's original algorithm to get the best overall performance.

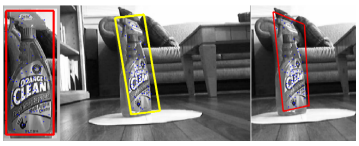
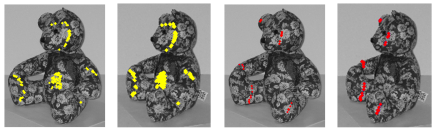


Figure 1: Simply replacing an affine fit (yellow) with a homography fit (red) improves recognition when out-of-plane rotation is high.



(a) $\mathbf{H_1}$ (b) $\mathbf{H_1}$ (c) $\mathbf{H_2}$ (d) $\mathbf{H_2}$

Figure 2: Homographies $\mathbf{H_1}$ and $\mathbf{H_2}$ need not correspond to object surfaces. Here, our method found two virtual planes that transect a rounded toy bear.

Provost and Fawcett [10] provide a principled answer to this question. They show that, when ROC data are available for several classifiers, classification quality can be maximized to be equivalent to the convex hull of all classifiers. The strategy for achieving this classification quality is based on the insight that the tangent line to the ROC curve can be expressed as

$$m = \frac{C_{fp} * NEG}{C_{fn} * POS}, \tag{8}$$

where,
m is the slope of a line in (FPR, TPR) space,
$\frac{NEG}{POS}$ is the ratio of true negatives to true positives,
C_{fp} is the cost for a false positive, and
C_{fn} is the cost for a false negative.

It follows that the optimal classifier in a particular use context is the one that lies on the convex hull at the point where the convex hull's tangent has the value computed by Equation 8. When the tangent touches two classifiers on the convex hull, the optimal strategy is to choose one of these two classifiers by a weighted coin toss. The weight given to each classifier is equal to its distance from the point on the convex hull that has the target FPR value.

In practice, the values on the right-hand side of Equation 8 are often unknown. In that case, the convex hull still serves as a guide for combining classifiers, and the choice for which classifier to use in a given context can be determined empirically.

4 Testing

4.1 Testing With SLAM Images

4.1.1 Experimental Setup

We evaluated SIFT matching, with and without our extension to handle parallax effects, on the task of scene matching in an indoor (home) setting. Training images consisted of six partially overlapping views, shown in Figure 3. Test images consisted of 454 views of the the same environment. 358 of these overlapped with one or more training views, and 97 contained views of areas that were not covered by the training images. The images

Figure 3: Training images for the visual SLAM experiment.

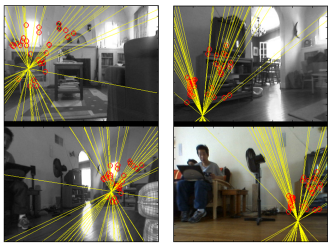
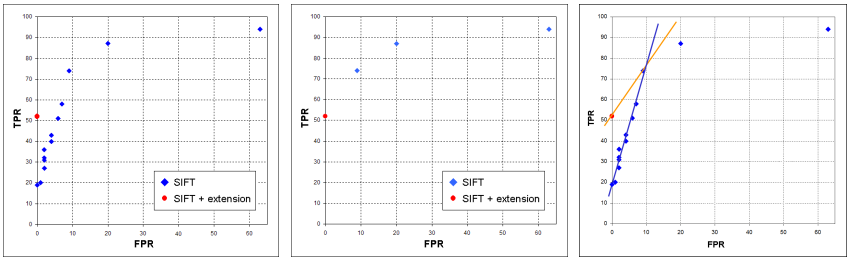


Figure 4: Example scene matches.

consist of video frames from a webcam mounted on a robot. The robot was driven, by remote control, through a house. The training views in these tests consist of frames taken near the beginning of the sequence.



(a) All ROC points (b) The convex hull (c) Change in ROC tangent

Figure 5: The ROC data for scene recognition.

4.1.2 Results

To analyze results from the SLAM tests, we pose a binary classification problem. The two classes are "known" and "unknown" location. The use of several, potentially overlapping, images to represent the "known location" class is then analogous to representing a multimodal distribution with one training sample per mode. Figure 5 shows ROC data for classification with and without 3D constraints. As this figure shows, at FPR=0, the ROC point for recognition with our extension lies well above the ROC point for Lowe's original method (TPR=52 with extension versus TPR=19 without).

From Equation 8, when $\frac{C_{fp}*NEG}{C_{fn}*POS} \geq (\frac{74-52}{9-0} = 2.4)$, our extension is the best classifier. Only when this ratio drops below $\frac{87-74}{20-9} = 1.2$ does Lowe's original method become the better choice. Between 2.4 and 1.2, choosing between these methods by a weighted coin toss is a better strategy than relying exclusively on either one.

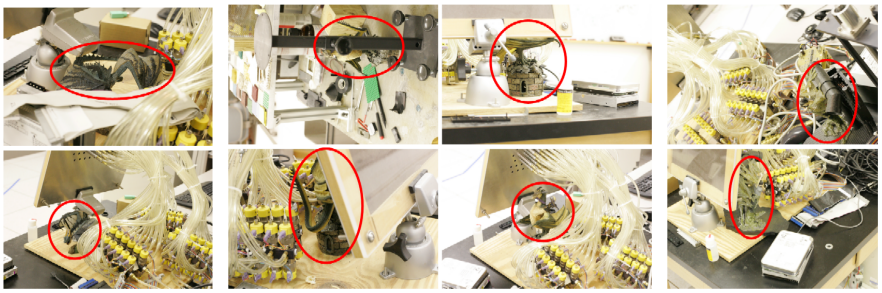
In addition to scene matching, using our extension gives 3D information. Figure 4 shows the epipolar geometry for example scene matches.

4.2 3D Object Recognition

4.2.1 Experimental Setup

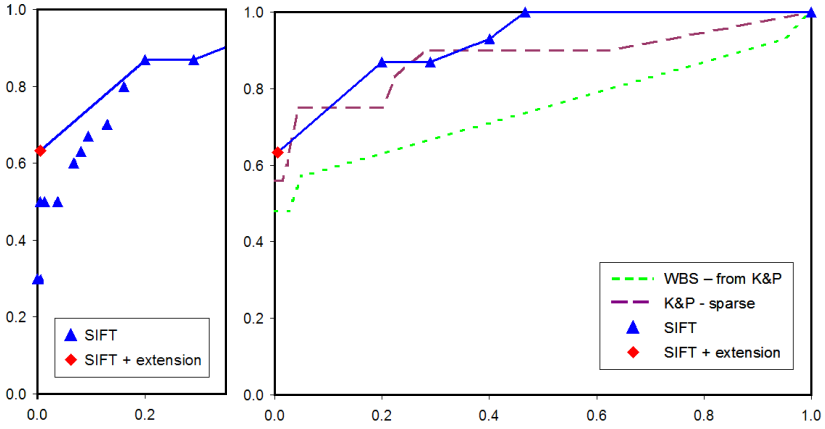
We also tested our extension to Lowe's method on 3D object recognition using the "geometrically complex" objects (two dragons and the chest buster) from the dataset that Kushal and Ponce used to evaluate their 3D recognition method [5]. The test images in this dataset are very challenging, consisting of 3D objects in various poses amidst dense clutter. In many, the objects are more than half obscured by feature-rich clutter.

Kushal and Ponce make use of 14-24 training images (seven-twelve stereo pairs) for each object. Although we could also have used all training images, we chose instead to use a small subset – less than 1/5 of their training images – since that better reflects how we expect our method to be used. We tested against all 80 test images.



(a) Small dragon (b) Green dragon (c) Chest buster

Figure 6: Examples of True Positives for 3D object recognition with our extension.



(a) Convex hull (b) Comparisons

Figure 7: ROC data for 3D object recognition tests.

4.2.2 Results

Figure 7 shows the ROC curve for the 3D object recognition tests. As 7a shows, at FPR=0, the ROC point for recognition with our extension again lies well above the ROC point for Lowe's original method (TPR=63 with extension versus TPR=30 without).

In [5], Kushal and Ponce follow a sparse, feature-based matching step with a dense, region-growing step to achieve very good results. Ours is a sparse method, which could also be followed by region growing. It's therefore interesting to compare our sparse method with theirs. They used DoG and Harris corner features, color histograms, and affine-rectified SIFT descriptors. We used only Lowe's original SIFT descriptors, and did not rely on color cues. As Figure 7b shows, recognition rates are similar, even though we used fewer than one fifth the number of training images. Both sparse methods noticeably outperform wide baseline stereo matching.

4.3 Consumer Products Dataset

4.3.1 Experimental Setup

Finally, we tested our extension to SIFT on Evolution Robotics Retail's consumer-products dataset. This dataset contains 255 training views of packaged consumer products, and 136 test images. Training images consist of up to six views of each product. A product barcode is associated with each training image. Test images were obtained under real world conditions. Ground truth for the test images consists of both the barcode and the most similar training image. The most visually similar training view for each test image was selected manually, prior to testing.

4.3.2 Results

Table 1 compares SIFT's performance with and without our extension. Two recognition results are reported for each method: Percent Correct View Matches and Percent Correct Product Matches. For View Matching, a correct match is scored only when the training view that is matched is the one that had been manually selected as being most visually similar to the test view. For Product Matching, a match is scored as correct if test and training images are linked to the same barcode, regardless of which view was matched.

5 Summary

Our extension to Lowe's method is efficient. The first homography, $\mathbf{H_1}$, adds zero overhead, since it directly replaces Lowe's step of computing an Affine warp by iterative least squares fit. To find $\mathbf{H_2}$ (or to give up on doing so) requires examining one or more additional Hough bins. However, Lowe's original algorithm also requires fitting transformations to multiple bins. With our extension, Lowe's method can be made significantly more robust to false positives, and the extension is easily toggled on and off as needed.

Acknowledgments

This work was funded by the US Office of Naval Research, Contract N00014-06-C-0033.

Method	View Matching	Product Matching
SIFT	44	89
SIFT + extension	68	97

Table 1: Percent correct matches on the consumer product dataset.

References

[1] V. Ferrari, T. Tuytelaars, and L. Van Gool. Simultaneous object recognition and segmentation by image exploration. *European Conference on Computer Vision, Proceedings*, pages 40–54, 2004.

[2] L. Van Gool and A. Zisserman. Grouping and invariants using planar homologies. *Workshop on Geometrical Modeling and Invariants for Computer Vision*, 1995.

[3] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[4] N. Karlsson, E. di Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, and M. E. Munich. The vslam algorithm for robust localization and mapping. *IEEE International Conference on Robotics and Automation, Proceedings*, pages 24–29, 2005.

[5] A. Kushal and J. Ponce. Modeling 3d objects from stereo views and recognizing them in photographs. *European Conference on Computer Vision, Proceedings*, 2006.

[6] D. G. Lowe. Object recognition from local scale-invariant features. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 2, 1999.

[7] D. G. Lowe. Local feature view clustering for 3d object recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[8] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. *European Conference on Computer Vision, Proceedings*, 1:128–142, 2002.

[9] H. Murase and S. K. Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision*, 14(1):5–24, 1995.

[10] F. Provost and T. Fawcett. Robust classification for imprecise environments. *Machine Learning*, 42(3):203–231, 2001.

[11] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. 3d object modeling and recognition using affine-invariant patches and multi-view spatial constraints. *International Journal of Computer Vision*, 66(3):231–259, 2006.

[12] C. A. Rothwell, A. Zisserman, D. A. Forsyth, and J. L. Mundy. Planar object recognition using projective shape representation. *International Journal of Computer Vision*, 16(1):57–99, 1995.

An Automatic Framework for Figure-Ground Segmentation in Cluttered Backgrounds

Leandro A. Loss, George Bebis, Mircea Nicolescu
Computer Vision Laboratory
University of Nevada, Reno
{loss,bebis,mircea}@cse.unr.edu

Alexei Skurikhin
Space and Remote Sensing Sciences
Los Alamos National Laboratory
alexei@lanl.gov

Abstract

This paper proposes an automatic framework for figure-ground segmentation of edged images in the presence of cluttered background. Our work employs perceptual grouping concepts to characterize image segments by means of their saliency, which is computed via tensor voting. The main innovation of our work is a case-based thresholding scheme which iteratively eliminates edge segments with low-saliency in multiple scales, preserving those that are more likely to belong to foreground. The key idea is classifying saliency histograms in several cases by considering the relative position of the modes of the figure/ground distributions and applying specific actions in each case. We have performed extensive experiments in order to evaluate our framework both quantitatively and qualitatively, including real images from the Berkeley dataset.

1 Introduction

Perceptual grouping can be defined as the ability to detect organized structures or patterns in the presence of missing and noisy information. Using local and low-level relations, such as proximity, smoothness, continuity etc. (i.e., Gestalt principles of perceptual organization), it can reveal important information about the global organization of structures in an image. A measure of structural organization, usually referred as *saliency*, can then be used as a discriminative feature to classify image elements as figure or background, therefore, supporting figure-ground segmentation.

Several important approaches have been reported in the past including [3], [7], and [8],. Recently, Williams and Thornber [9] have proposed a probabilistic approach based on closed random walks, where saliency is defined relatively to the number of times an edge is visited by a particle in a random walk. The use of a voting process to infer salient structures from noisy and sparse data was introduced by Guy and Medioni [2] and then formalized into a unified tensor voting framework [6]. Tensor voting represents input data as tensors and interrelates them through voting fields built from a saliency function that

incorporates the Gestalt principles of proximity and good continuation. In the past, we have employed tensor voting in order to characterize the salience of edge segments [4]. In particular, we performed figure-ground segmentation by analyzing saliency information at multiple scales and removing low-saliency segments in an iterative fashion. This approach has shown to produce good results, however, its success relies on several parameters that were chosen manually.

In this paper, we propose a new approach for figure-ground segmentation which relies on perceptual grouping of features embedded in the tensor voting framework. Following Loss et al. [4], we perform figure-ground segmentation by analyzing the saliency of edge segments at multiple scales and removing low-salience segments using an iterative scheme. The main contribution of our work is automating the above framework by introducing a case-based thresholding scheme to eliminate low-saliency segments more effectively and criteria for stopping the iterative voting without user intervention.

Our method works by first decomposing the image edges into directional tensors and applying tensor voting in multiple scales. Then, histograms of segment saliencies are built for each scale and their modes are detected using Mean Shift (MS) [1], an adaptive gradient ascent method. Next, a case-based thresholding methodology is applied which categorizes saliency histograms into six general cases. This is performed using information about the relative position of the modes of the figure/ground distributions in the saliency histogram. Finally, low-salience image edges are eliminated according to the case detected. The whole process keeps repeating until certain stopping criteria are met.

The proposed method has several advantages including that: (1) there is no need to assume a global model for the structures present in an image; (2) it imposes no *a-priori* assumptions on the scale of the objects present in an image or their number; (3) it does not assume any distribution model for the saliency histograms; (4) thresholding low-saliency edge segments is done in an automatic fashion; (5) it yields results that are competitive with state-of-the-art methods; and (6) it can provide higher quality input to attentional methods, object detectors and recognizers.

2 Segmentation Using Tensor Voting

In the framework proposed by Guy and Medioni [2], the input data is encoded as elementary tensors. Supporting information, including proximity, smoothness and continuity, is propagated from tensor to tensor by vote casting. Tensors that lie on salient features, such as curves in 2D, or curves and surfaces in 3D, strongly support each other and deform according to the prevailing orientation, producing generic tensors. Each such tensor encodes the local orientation of features (i.e., given by the tensor orientation), and their saliency (i.e., given by the tensor shape and size). Important features can be then extracted by examining the tensors resulting after voting. The method is robust to considerable amounts of outlier noise and does not depend on critical thresholds. The only free parameter is the scale factor σ, which defines the voting fields. Specific details can be found in [2] and [4]

Although the scale factor σ of the tensor voting framework has been shown to be fairly stable [6], it is subjected to the same trade-off of every scale-dependent method: small scales capture local structures while large ones capture global configurations. In real scenarios, however, structures emerge from different scales and the prediction of the optimal scale is fairly complicated. Moreover, it is impossible in general to choose a

fixed threshold that would provide a good figure segmentation due to the complexity and amount of background in images. We have addressed both issues in [4] by introducing a multi-scale scheme that removes background segments conservatively in an iterative fashion, leading to an improved figure-ground segmentation.

3 Automatic Framework

Like in [4], we eliminate image edges iteratively by analyzing their behavior at multiple scales, however, the main advantage of our method is that it works in an automatic fashion. To halt the iterative process automatically but also to determine at each iteration what action is more effective for a certain type of histogram, we have devised a methodology that categorizes saliency histograms in different cases. This is done by detecting the modes of the figure/ground distributions in the saliency histogram using MS and analyzing the relative position of their peaks. If the figure/ground distributions are multimodal, then we only consider the nearest pair of figure-ground modes for classification purposes.

Considering the relative position between figure/ground modes in the saliency histogram provides strong information about the amount of overlap between figure and ground distributions. When the modes are positioned close together, then the figure distribution overlaps more with the ground distribution making segmentation hard. On the other hand, when the distance is large, then the amount of overlap is smaller making segmentation much easier. Our objective is to distinguish among these cases and apply specific actions in each case in order to automate the iterative tensor voting scheme but also make the segmentation process more effective (e.g., avoid using fixed thresholds, minimize iterations, and possibly improve segmentation results).

The histogram categories were determined during a training phase by clustering a large number of saliency histograms using MS (i.e., MS is used twice in this work; first, to detect the modes of a saliency histogram and second, to cluster saliency histograms). The saliency histograms used for training were computed from a well known dataset which contains real objects (i.e., fruits and vegetables) in textured backgrounds [9]. Using this dataset rather than a collection of various images for training allowed us to consider a much larger space of possible configurations between figure and ground distributions. We have experimentally verified that saliency histograms obtained from quite different images do resemble saliency histograms obtained from the dataset used for training (see Section 4.3).

Applying MS on the dataset of [9] yielded six clusters as shown in Fig. 1(a). Based on these results, we classify saliency histograms in six cases as shown in Fig. 1(a) (i.e., cases have been labeled for easy reference). The only parameter involved in determining the histogram categories is MS's bandwidth which was set to 0.20 both for the detection of the histogram modes as well for clustering of the histograms into cases. The cases were noted to be immune to small changes of this parameter. Depending on the case that the saliency histogram of a new image falls to, we apply one of three different actions. These actions were formulated carefully in order to preserve the conservativeness criterion proposed in [4], but also to take advantage of well separated distributions and avoid unnecessary iterations. At a given iteration, a specific case is applied if and only if this case appears in at least 50% of the image's saliency histograms (i.e., corresponding to different scales), otherwise, Case 6 is chosen for its respective more conservative action.

Case 1 is the simplest and most well-behaved case. It represents a very low salience peak (i.e., background) next to one or more highly salient peaks (i.e., figure). This case usually appears in high SNR, non-salient background, salient figure images. In this case, the action taken is a direct clustering of the elements. Clustering is performed using MS's gradient information to determine what peak each element belongs to. To be eliminated, an element must appear in the same cluster in all Case-1 saliency histograms of the image. Cases 2 and 4 represent configurations where a very low salient peak is detected, but a second peak is either detected close to the first or not detected at all. These cases are present in low SNR, non-salient background, low salient figure images. In this case, thresholding is applied at the first mode position, and an element eliminated if it is below the threshold in all Case-2 or Case-4 saliency histograms of the image. Finally, Cases 3, 5 and 6 represent configurations where no low salient peak is detected. In this case, thresholding is performed at a very low value in order to generate some change in the image for the next iteration, preserving figure integrity. Again, an element is eliminated if it is below the threshold in all Case-3, Case-5 or Case-6 saliency histograms of the image. This last action represents the implementation of [4]'s methodology.

Thresholding actions have the objective of reducing image complexity and, hopefully, bring the histogram configuration to Case 1. There are three stopping criteria based on the changes that occur in various cases. Specifically, the algorithm converges when a Case-1 saliency histogram changes to any other case. This criterion takes into consideration that once the peaks are enough far apart, any attempt to switch the configuration to a more complex one might imply that the background was successfully eliminated. The most common transitions from Case 1 are to Case 3 (i.e., no background left, figures of different saliency intensity create multiple peaks) or Case 5 (i.e., no background left, figure populates the whole saliency spectrum).

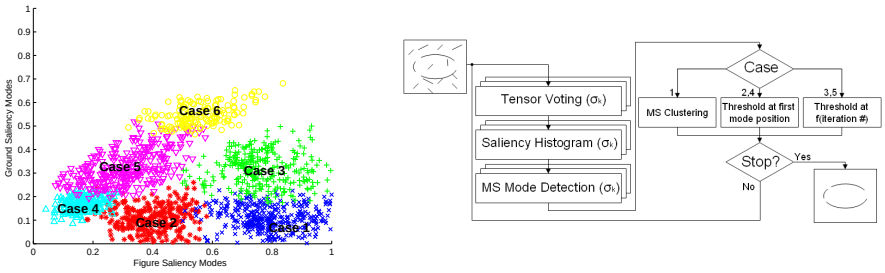
The number of elements eliminated is another criterion used to stop the algorithm. As mentioned before, MS clustering is performed if Case 1 is detected, and elements are eliminated if they belong to the least salient cluster in all scales categorized as this case. Therefore, differently from the others, this is the only case where there might be no elimination of elements. If this happens, the algorithm is assumed to have converged and stops. This captures the cases where small peaks are formed close to the low saliency region, but there is no element weak enough to be there for more than a few scales.

The third stopping criterion covers configurations that never converge to Case 1. In this case, the algorithm will keep eliminating parts of the image without stopping. Instead of choosing a maximum number of iterations, we adapt the threshold of Cases 3, 5 and 6 proportionally to the iteration index (i.e., similarly to [4]'s methodology), and assume that the algorithm has converged if this value exceeds the boundary between Cases 3 and 5. Specifically, we adapt the threshold value according to the expression $T_i = 0.06 + i \times 0.02$, where i stands for the iteration index. Fig. 1(b) shows the block diagram of our automatic framework.

4 Experimental Results

4.1 Experiment 1: Tests on Fruits and Textures Dataset

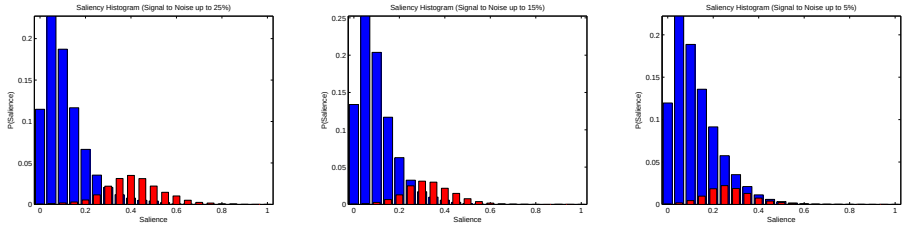
First, we experimented with the set of fruit and texture sampled silhouettes, used in [9] and [4]. This dataset contains real objects (i.e., fruits and vegetables) in textured backgrounds



(a) (b)

Figure 1: (a) The six cases obtained during training by applying MS clustering on pairs of figure-ground modes detected from a large number of saliency histograms. (b) Block diagram of our automatic framework.

(i.e., leaves, rocks, etc). As in [9] and [4], each benchmark image was built from a pair of sampled silhouettes belonging to a fruit or a vegetable and textured background. Nine figure silhouettes were re-scaled to an absolute size of 32x32 and placed in the middle of nine 64x64 re-scaled ground windows. In addition, 5 different signal to noise ratios reduced the number of ground segments proportionally to the number of figure segments. This dataset has a total of 405 images (9 figures and 9 backgrounds at 5 different SNR). Fig. 3(a)-(c) show some examples of the benchmark images at different SNR. Fig. 2 shows plots of saliency histograms. The red bars correspond to figure segments while the blue ones to background segments. Note that traditional approaches would fail in determining a threshold value that could generate a good figure-ground segmentation, especially for SNR lower than 20% (see [4]'s Fig. 3).



(a) SNR at 25% (b) SNR at 15% (c) SNR at 5%

Figure 2: Saliency histograms according to SNR (red bars - figure, blue bars - ground). As SNR decreases, the overlap between the curves belonging to ground and figure increases.

Fig. 5(a) shows the results of applying our framework on the fruit and texture dataset. Each curve reveals the behavior of our approach as a function of SNR. Each curve represents average results over all the images in the dataset and each dot corresponds to the result of one iteration. Curves with fewer dots converged faster to the final result. Fig. 3 shows some visual results.

In summary, our framework was able to eliminate more than 90% of the background, preserving on average more than 85% of the figure. For images with up to 5 times more

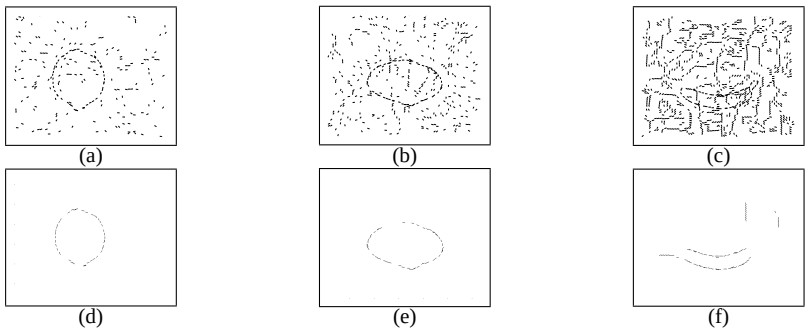
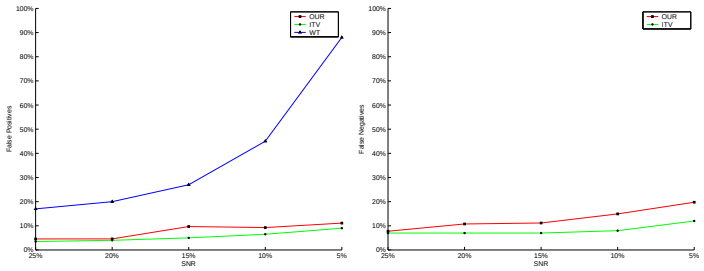


Figure 3: Visual results on fruit and texture dataset. (a) Lemon on brick with SNR at 25%, (b) red onion on leaves with SNR at 15%, (c) banana on bark with SNR at 5%, (d) lemon on brick upon 3 iterations, (e) red onion on leaves upon 6 iterations, (f) banana on bark upon 8 iterations.

background elements than figure segments, our framework eliminated more than 95% of the background, preserving more than 90% of the foreground. Highly noisy images tend to converge slower (i.e., 8 iterations) but the results obtained were very good considering the nature and conditions of them. Traditional approaches have much worse performance on this kind of images due to the large overlap between background and figure distributions (see Fig. 2c).

We have also performed comparisons between the proposed framework (OUR) and the methods proposed by Williams and Thornber (WT) [9] and Loss et al. (ITV) [4] considering the same fruit and texture dataset . The methods were compared based on the False Positive (FP) and False Negative (FN) rates according to the Signal-to-Noise Ratio (SNR). Fig. 4 shows both FP versus SNR and FN versus SNR curves. No curve is shown for WT in the case of FN versus SNR because no false negative rates were provided in [9]. ITV's curve shows results from the iteration that produced the best outcome, as if an optimal stopping criterion existed for that method.



(a) FP vs SNR (b) FN vs SNR

Figure 4: Comparison with [9] (WT) and [4] (ITV) methods.

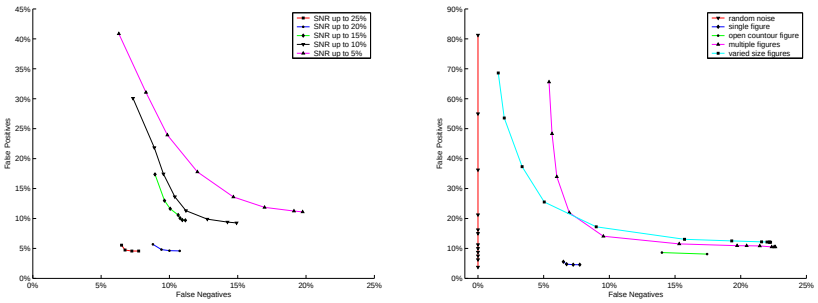
This result shows that our framework is very competitive with ITV (on average 2% worse in terms of FP and 4% in terms of FN), and in the simplest case, 12% better than

WT. Differently though from ITV, our method is fully automatic by taking decisions based on a set of predetermined cases. Our method and ITV are shown to deal better than WT with highly noisy images. This can be seen by the quasi-linear behavior of the curves corresponding to these methods. This is mainly due to the characteristic of these frameworks that eliminate elements iteratively, reducing the complexity of the image step by step.

4.2 Experiment 2: Tests on Extended Fruit and Textures Dataset

The fruit and texture dataset offers a good means of experimentation and comparison. It is composed by real images, that approximate real applications in computer vision. It is also important to notice that the background is well organized. However, since the figures are always from closed objects and displayed in the same position and scale, the benchmark lacks challenges of this sort. We have extended this benchmark by using the same fruits and textures, but incorporating new characteristics that make it more challenging and complete. In particular, we have created more test images by varying the number of fruits and their sizes, and by removing sequences of fruit segments, opening their silhouettes. We also included a set of images composed by a circle and random noise for the sake of comparison.

In summary, four new datasets were created: (1) open contours - 405 images; (2) multiple objects - 810 images; (3) varied-size figures - 810 images (4) random noise background - 200 images. Fig. 6(a,b,c) shows some examples from these new datasets. The objective of these extra datasets is to show that our method does not rely on the closure of a contour and does not make any assumptions on the size of the objects present in an image. Fig. 5(b) shows the results of our framework for each dataset. The SNR of all the images in this experiment was set to 25%. Fig. 6 shows some visual results.



(a) (b)

Figure 5: (a) Results on the fruit and texture dataset. (b) Comparison of our framework using different datasets.

These results illustrate that our framework has no bias toward contour closure, size or nature of foreground. On average, our framework eliminated more than 90% of the background, preserving more than 85% of the foreground. The higher number of iterations needed for multiple and varied-size figures is due to the higher number of background elements in those images. In the case of backgrounds made of random noise, our system eliminates more than 97% of the background elements, preserving 100% of foreground. Still, most of the background preserved lies close and aligned to the figure segments.

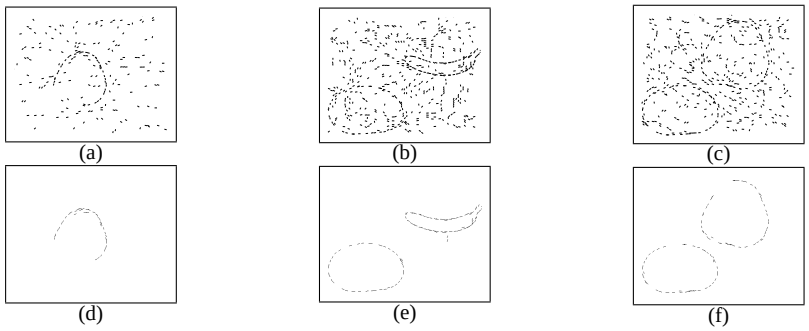


Figure 6: Visual results on extended datasets with SNR at 25%. (a) Incomplete pear on water, (b) avocado and banana on bark, (c) avocado and larger pear on fabric, (d) incomplete pear on water upon 2 iterations, (e) avocado and banana on bark upon 7 iterations, (f) avocado and larger pear on fabric upon 7 iterations.

4.3 Experiment 3: Tests on Real Images

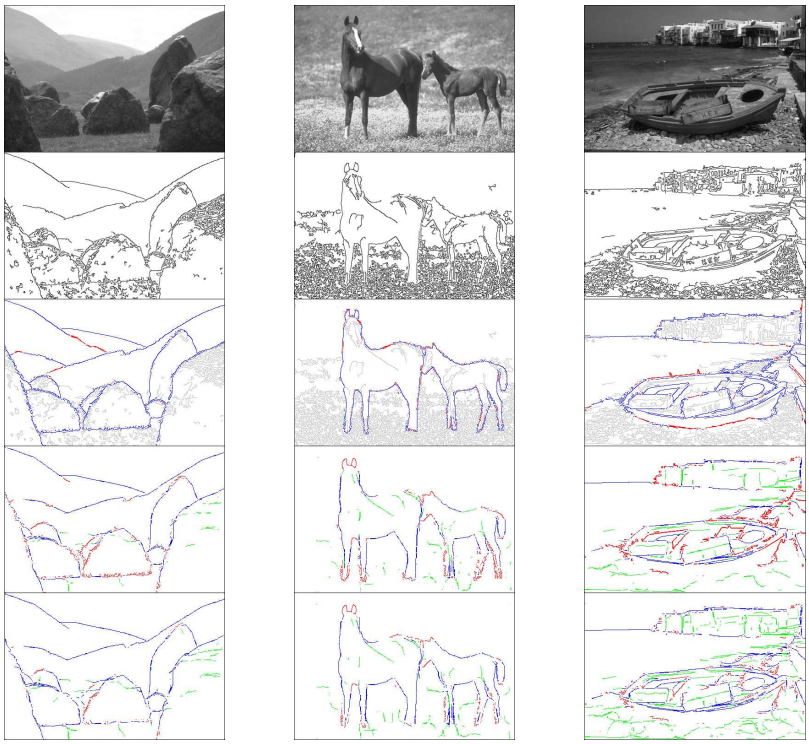
Finally, we tested our framework on real images taken from the Berkeley Segmentation Dataset [5]. This dataset contains ground truth information obtained by asking several human subjects to identify the most important segments in each image. For consistency, we considered only those segments found by two or more human subjects. Fig. 7(top) shows three examples. Since our framework works with segments, we have pre-processed each image using the Canny edge detector (Fig. 7 (second from the top)). Edges smaller than 10 pixels were removed.

To assess the results of our method, we chose the parameters of the Canny edge detector such that it preserved as many segments present in the ground truth as possible. This is because our current implementation of tensor voting can not account for missing edges by filling the gaps between edges. Therefore, if a ground truth edge is not among the edges found by the Canny edge detector, it cannot be found by our method. Obviously, this assumption can be relaxed by employing a more powerful version of tensor voting called ”dense tensor voting” [6] which can fill in gaps between edges.

Fig. 7 (third from the top) shows the ground truth segments superimposed on the edge image. For visualization purposes, blue represents segments present both in the ground truth and edge image, red represents segments present in the ground truth but not in the edge image, and gray represents edges present in the edge image but not in the ground truth. For evaluation purposes, we are interested in maximizing the number of blue segments that our method can keep while minimizing the number of gray pixels. Red pixels cannot be detected by our method since they are not present at all in the input to our algorithm.

Figure 7(bottom two) shows the resulting salient segments found by ITV and OUR methods. In this case, all blue segments correspond to True Positives (TP), all red segments correspond to False Negatives (FN), while all green segments correspond to False Positives (FP). TPs were computed by comparing the salient segments found by our method to those present in the ground truth segments which were also present in the edge image. FNs were the ground truth segments present in the edge image that were eliminated by our method. Finally, FPs were salient edges kept by our method that were

not present in the ground truth. Table 1 shows specific results for each of the images shown in Fig. 7.



(a) Test Image #36 (b) Training Image #146 (c) Training Image #180

Figure 7: Images from the Berkeley segmentation dataset. From the Top: original image, Canny edge image, Ground truth superimposed over the edge image, result of ITV, result of OUR method. Refer to text for color scheme.

These results reveal the high accuracy of our method in finding meaningful segments in an image. It should be noted that FPs should not be regarded as errors since most of them correspond to edges belonging to some meaningful salient curves. The results using the Berkeley dataset are very encouraging since although our framework was designed mainly using synthetic data, it generalizes very well on real images.

5 Conclusions and Future Work

We have proposed an automatic framework for figure-ground segmentation based on an iterative, multi-scale tensor voting scheme. Our experimental results and comparisons indicate that our framework is very competitive with state-of-the-art approaches, with the advantage of being automatic. The key to automating the segmentation framework was the case-based thresholding methodology proposed here. For future work, we plan to formalize our case-based thresholding into a reasoning framework by employing fuzzy

Table 1: Segmentation results using images from Berkeley dataset [5].

	ITV			OUR		
Berkeley Image	TP	FN	FP	TP	FN	FP
Test #36	55%	29%	5%	88%	12%	18%
Training #146	57%	43%	6%	76%	24%	6%
Training #180	38%	46%	13%	75%	25%	29%

rules. Moreover, we plan to improve the time requirements of iterative voting scheme by updating the votes at each iteration instead of recomputing them from scratch.

References

[1] K. Fukunage and L.D. Hostetler. The estimation of the gradient of a density function, with application in pattern recognition. *IEEE Trans. Information Theory*, 21:32–40, 1975.

[2] G. Guy and G. Medioni. Inference of surfaces, 3-d curves, and junctions from sparse, noisy 3-d data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11):1265–1277, 1997.

[3] L. Hérault and R. Horaud. Figure-ground discrimination: A combinatorial optimization approach. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 15:899–914, 1993.

[4] L.A. Loss, G. Bebis, M. Nicolescu, and A. Skourikhine. Perceptual grouping based on iterative multi-scale tensor voting. *Lecture Notes in Computer Science 4292 - ISVC'06*, 2:1786–1797, 2006.

[5] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[6] G. Medioni, M.-S. Lee, and C.-K. Tang. *A Computational Framework for Segmentation and Grouping*. Elsevier Science, 2000.

[7] S. Sarkar and K. Boyer. Quantitative measures of change based on feature organization: Eigenvalues and eigenvectors. *Proc. IEEE Conference on Computer Vision and Pattern Recognition - CVPR'96*, pages 478–483, 1996.

[8] S. Ullman and A. Sha'ashua. Structural saliency: The detection of globally salient structures using a locally connect network. *2nd. International Conference on Computer Vision - ICCV'88*, 1988.

[9] L. Williams and K.K. Thornber. A comparison of measures for detecting natural shapes in cluttered background. *International Journal of Computer Vision*, 34(2/3):81–96, 2000.

A framework for learning to recognize and segment object classes using weakly supervised training data

Caroline Pantofaru and Martial Hebert
The Robotics Institute, Carnegie Mellon University
Pittsburgh, Pennsylvania, USA
crp@ri.cmu.edu, hebert@ri.cmu.edu

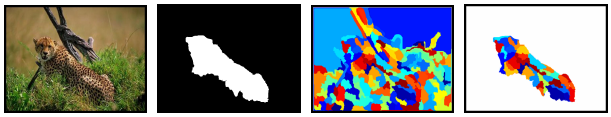
Abstract

The continual improvement of object recognition systems has resulted in an increased demand for their application to problems which require an exact pixel-level object segmentation. In this paper, we illustrate an example of an object class recognition and segmentation system which is trained using weakly supervised training data, with the goal of examining the influence that different model choices can have on its performance. In order to achieve pixel-level labeling for rigid and deformable objects, we employ regions generated by unsupervised segmentation as the spatial support for our image features, and explore model selection issues related to their representation. Numerical results for pixel-level accuracy are presented on two challenging and varied datasets.

1 Introduction

In this paper, we illustrate an example of an object class recognition system which is trained using weakly supervised training data, and the influence that different model choices can have on its performance. The primary objectives of our system are to *recognize* and *segment* object classes in *novel images* using models learned from *weakly supervised* training data for which image-level labels exist but no indication of individual object localization exists. In order to achieve pixel-level labeling such as in Figure 1 for rigid and deformable objects, we employ regions generated by unsupervised segmentation as the spatial support for our image features. By using established methods for weakly supervised learning, we can concentrate on the actual instantiations of the region representations and examine how different choices for these instantiations affect performance.

Performing pixel-level recognition and segmentation of learned object classes in novel images is itself a very difficult problem which requires image classification to determine the presence of the object, identification of discriminative object parts to roughly localize the object, and identification of non-discriminative object parts to complete the segmentation. For object recognition, local interest point-based methods which use highly discriminative features such as SIFT [17] on fixed-shape object patches have been very successful. These methods do not, however, provide a way to exactly determine which pixels are part of the object unless the shape of the object is rigid or can be parameterized or modelled easily. So we join a growing number of approaches which advocate using regions from unsupervised segmentations to group pixels together and allow proper classification of object parts which are not themselves discriminative [30, 22, 23, 25, 14, 16, 9, 10].



(a) (b) (c) (d)

Figure 1: For the image of the cheetah in (a), our goal is to produce an exact, pixel-level object segmentation mask as in (b). Regions generated by unsupervised segmentation (c) can be classified to produce such a mask (d).

Many systems which attempt to identify all of the pixels in an object require fully supervised training data that has the objects segmented or extremely obvious [29, 15, 9, 16, 31, 26, 2, 18, 32, 8, 11, 21, 10, 28]. Training data that contains pixel level segmentations is extremely tedious to obtain, however, so such systems are not able to scale to large datasets. On the other hand, object discovery performed in large datasets of completely unlabeled images, such as in [25] can lead to the discovery of objects which are unintuitive or unimportant. Additional constraints may also be needed, for example that an object of interest be exactly denoted by one region [25] or that video data be available [14]. A compromise is to have weakly labeled data in which images are labeled with the objects of interest that they contain, but without the object locations. This type of data is far cheaper to obtain, and has been used in numerous image classification and object recognition tasks that utilize interest points (sparse or dense) or fixed-shape image patches, such as [22]. Very few systems, however, have attempted to use weakly supervised training data and regions to learn object segmentations and object classifiers [1, 6, 3].

As a consequence of using weakly supervised training data and requiring pixel-level object segmentation as output, the image features employed must be carefully designed. As discussed, we cannot use interest points or fixed-shape patches, so we choose to use region-based features. Since we do not know the object shape a priori, we cannot use it as a cue as done in [29, 14, 16]. As argued by [24], we also cannot assume that one segmentation region will correspond to an entire object of interest. So we use and compare three region descriptors which are based on region texture and discriminative region features. Two of the features are adaptations of the texton histograms introduced in [19] to regions, and the third is the Region-based Context Feature (RCF) [23], which includes information about discriminative image features in and around a region. Although other representations are possible, by instantiating our algorithm with these three concrete representations we can study how model changes affect performance.

The structure of the remainder of the paper is as follows. We begin by discussing our system framework which includes our image segmentation method and the learning algorithm used for classifying regions. We then describe the datasets on which results will be presented. With this general framework in mind, we classify single regions using the various representations and compare their performance. We then examine methods for combining representations. Finally, we examine whether incorporating information from neighbouring regions allows for more robust detection and localization.

2 Framework

The general framework for all of the object recognition and segmentation experiments in this paper is as follows: images are first divided into regions using unsupervised seg-

mentation, a representation is then computed for the regions, a model is learned which determines whether a given region feature is indicative of the object in question, and finally novel regions are classified to produce object masks in novel images. This section lays the groundwork for the presented experiments by briefly describing the segmentation and classification methods used, as well as the datasets.

2.1 Segmentation

Although our approach is independent of the segmentation algorithm, it is useful to understand the nature of the regions we are working with. We use mean shift-based segmentation [4] with pixel features of image position (2-dimensional), L*u*v* colour (3-dimensional), and a texton histogram (30-dimensional). An example segmentation is given in Figure 1. We use the algorithm from the Berkeley segmentation database website [19, 20] to compute the texton histograms; the texton at each pixel is a vector of responses to 24 filters quantized into 30 textons, and the texton histogram centred at a pixel is an accumulation of the textons in a 19x19 pixel window. The low dimensionality of our texton histograms allows for generalization during segmentation, grouping together pixels surrounded by similar but not identical textures. Some scale invariance is achieved by segmenting each image at three different image sizes with the same parameters. For clarity, we derive the formalism in this paper for only one segmentation of each image.

2.2 Classification model

Once we have segmented the image into regions, we can represent each region by a feature F (defined in Section 3). To perform region classification, we require a score for each region-representing feature. The method we use for scoring features in our weakly supervised learning environment was introduced for interest points in [5, 27], and for regions in [23]. We assign a score to each feature which indicates how well it discriminates between the positive (object) and negative (background) classes based on the image labels. Let O indicate the *presence* of the positive object class in an *image*, and $\overline{O}$ the *absence* of the object class in an *image*. Then we can define R as the log likelihood ratio of the object's presence, and score $\tilde{R}$ as our posterior belief in O given F (assuming that $P(O) = P(\overline{O})$):

$$R(F) = \log \frac{P(F|O)}{P(F|\overline{O})} \in (-\infty, \infty) \qquad \tilde{R}(F) = P(O|F) = \frac{P(F|O)}{P(F|O) + P(F|\overline{O})} \in [0,1] \quad (1)$$

Note that $\tilde{R}$ preserves the ordering of R but rescales the scores to lie in $[0,1]$. For R, a score approaching negative infinity implies that F indicates a negative image, a score approaching infinity implies that F indicates a positive image, and a score of 0 indicates that F is uninformative for either class. For $\tilde{R}$, a score of 0 implies a negative image, a score of 1 implies a positive image, and a score of 0.5 is uninformative. Laplace smoothing is performed during numerical evaluation of the probabilities for robustness.

For a novel image, we generate its segmentation, compute a feature for each region, and classify each region based on the learned score $\tilde{R}$. The union of all positively classified pixels (or regions) is the object segmentation.

2.3 Datasets

Creating datasets for weakly supervised learning requires much less effort than fully segmenting all of the objects. There is a trade off, however, in that the images chosen for

the weakly supervised datasets must facilitate tractable learning. Specifically, the backgrounds in the images which contain the object must share features with the images which do not contain the object. In this way, we can learn that image features which are seen in both object and non-object images are in fact part of the background. In addition, since we compute pixel-level accuracy, we require images with segmented objects in the *test* image set (but *not* in the training image set).

We present results on two datasets. The first is the 'Spotted Cats' dataset used in [23], which consists of data from the Corel Image Database. The folder 'Cheetahs, Leopards and Jaguars' serves as the positive class, and the folder 'Backyard Wildlife' the negative class. Each folder contains 100 images, half of which are used for training, and half for testing. The images contain cats in a variety of poses, at a variety of sizes, and in varying numbers.

The second dataset we use is the 'Cars' dataset comprised of images from the PASCAL challenge VOC2006 dataset [7]. For training, we use the cars training and validation images (553 images) as the positive set. As the negative training set we use all of the bicycle and bus training and validation images which do not contain cars (509 images). For testing, we use the cars test set (544 images). The PASCAL VOC2006 dataset is extremely challenging, with a large variety of object sizes, poses, lighting conditions, and occlusion. A large number of the images contain multiple cars. We have produced ground truth object masks for the test set for the purpose of the experiments herein.

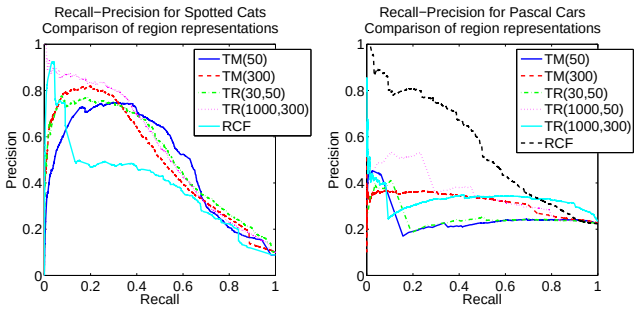
The results we produce are pixel-level, in other words they compute the recall and precision of classifying each individual pixel in each image as object or background. Note that most of the existing evaluations for recognition and localization (which do not rely on human segmentations) are presented as either image classification or object localization in terms of bounding boxes or object centres, with a few exceptions such as [23, 25, 29, 9]. Thus the pixel-level results we present are unusually strict. For most of the experiments, recall-precision curves are presented as in Figure 2. Given the volume of experiments, we summarize some of them in Table 1 by their average precision.

3 Region representations

In order to classify regions we require a way to represent them. In this section, we show that region representation can in fact strongly affect performance. We present three approaches to modeling regions, two based on histograms of textons [19], and another by the Region-based Context Feature [23], and evaluate their performance.

Texture representation 1: mode of the texton histograms in a region. Texture is a representation of the average, or repetitive, patterns within a region. For segmentation, we use a 30-dimensional texton histogram computed at each pixel in the image. Since our segmentation method is mean shift-based, and mean shift computes the mode of the features within a region, a texton histogram mode (TM) for each region results directly from the segmentation. The 30-dimensional texton histogram modes from every training region are clustered into a vocabulary of size $K_{TM} \in \{50, 300\}$, and new modes are assigned to the cluster with the closest centroid.

Texture representation 2: histogram of the textons in a region. As discussed in section 2.1, a low-dimensional texton space facilitates segmentation. However, for region classification a more discriminative texton vocabulary would lead to more discriminative descriptors of region texture. The second texture representation, then, requires computing a new texton vocabulary whose size, $K_{TEX} \in \{30, 200, 1000\}$, is independent of that used for segmentation. The texton words in a segmentation region are then accumulated to



(a) (b)

Figure 2: Comparison of region representations by pixel-level recall-precision curves for recognition and segmentation of (a) Spotted Cats vs. Backyard Wildlife, and (b) PASCAL Cars vs. background. TM means texture representation 1, the texton mode; TR means texture representation 2, the histogram of textons over the region; RCF means Region-based Context Feature. The first number is the size of the texton dictionary used, and the second is the size of the texton histogram mode or texton histogram dictionary.

form a new texton histogram (TR). Note that this texton histogram is computed over the region, not over square windows as in the previous method. These histograms can be clustered to create a different vocabulary of region descriptors of size $K_{TR} \in \{50, 300\}$.

Region-based Context Features. The shape of a segmentation region is data-driven, which is useful for specifying the spatial support of texture features, but it can also work against us. Consider the images in the top row of Figure 4(b,e). The regions on the body of the cheetah are detected using texture, but the head and face are missed since they have little interior texture. The head regions do, however, have discriminative features such as the shape of the ear or the presence of an eye. Cars also have little texture within a region, but the shape of the wheels are strong cues. For these image structures, we require a representation which can both encapsulate their uniqueness and extend past the boundaries of the region to capture shape. To address these concerns, we adopt the Region-based Context Feature (RCF) [23]. RCFs are the cluster centres of clustered histograms of quantized SIFT [17] descriptors computed at local interest points in and *around* a given region. The SIFT descriptors ensure their discriminative power, and the histograms include points outside the region in a principled manner. For implementation details, please see [23].

Experiments. We experiment with model selection for region representation by choosing a representation type, TM, TR or RCF, a texton or SIFT dictionary size K_{TEX}, and a size for the descriptor, or histogram, dictionary K_{TM}, K_{TR} or K_{RCF}. In Figure 2 and Table 1 we compare choices for these parameters. Figure 2 presents recall-precision curves for pixel-level detection on the two datasets. For each pixel in each image, we evaluate whether it has been assigned to the correct class. Curves are shown for the TM representation (texton representation 1) with $K_{TM} \in \{50, 300\}$, the TR representation (texton representation 2) with $(K_{TEX}, K_{TR}) \in \{(30, 50), (1000, 300)\}$, and the RCF representation with $K_{SIFT} = K_{RCF} = 50$. Curves corresponding to the other model choices are omitted for clarity. Table 1 presents values for the average precision of all model choices. The top row corresponds to the representation, the second row to the texton or SIFT dictionary size, and the left column to the feature dictionary size.

While larger dictionaries seem intuitively more desirable, the results in Figure 2 dis-

	TM 30	TR 30	TR 200	TR 1000	RCF 50
50	0.493	0.513	0.504	0.542	0.392
300	0.498	0.493	0.533	0.545	-

(a)

	TM 30	TR 30	TR 200	TR 1000	RCF 50
50	0.246	0.253	0.302	0.353	-
300	0.316	0.261	0.346	0.329	0.548

(b)

Table 1: Comparison of region representations by average precision computed on (a) the Spotted Cats dataset, and (b) the PASCAL cars dataset. The first row indicates the region representation used, the second row indicates the size of the texton (or SIFT) dictionary, and the left column indicates the size of the descriptor (histogram) dictionary. Note that the average precisions do not increase monotonically with dictionary size.

agree. For the TM representation on the Spotted Cats dataset, a dictionary size of 300 performs better for low recall values, a dictionary size of 50 performs better for medium recall values, and they perform similarly for high recall values. This trend can also be seen for the TR curves for the Spotted Cats. In Table 1 we can see that the average precision does not increase monotonically with dictionary size, reinforcing the fact that model selection is crucial to meeting specific system requirements.

4 Combining features

In the previous section, we described three different ways to represent segmentation-generated regions for object recognition and localization: TM, TR and RCFs. We found that the texture descriptors are better suited to objects with regular texture, such as spotted cats, while the RCFs are better suited to objects with discriminative parts, such as cars. This section focuses on combining the representations with the goal of eliminating the need to make a class-dependent choice between them.

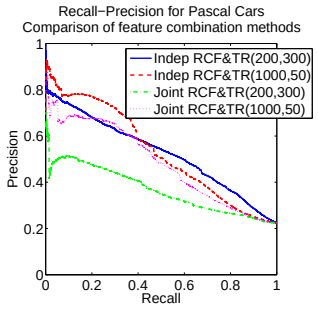


Figure 3: Recall-precision curves comparing feature combination methods for pixel-level recognition and segmentation of the PASCAL Cars. 'Indep' means combining features assuming independence; 'Joint' means using joint distributions.

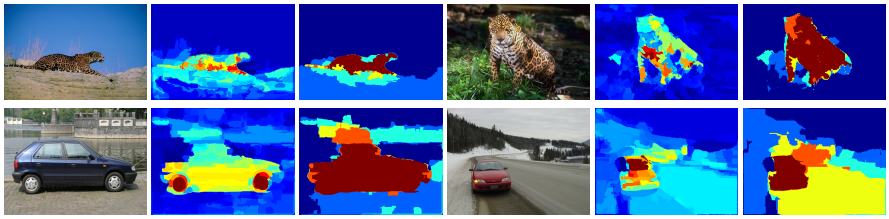
To combine feature sets, a simplifying assumption that could be made is feature independence. Although a texture representation of a region and an RCF are clearly not independent, this naïve Bayes assumption allows us to estimate their joint probability from relatively little data. We define the independent score for a feature pair to be:

$$\tilde{R}(T,RCF) \propto \tilde{R}(T)\,\tilde{R}(RCF)$$

Where T is a texture feature, and RCF is a Region-based Context Feature.

With sufficient training data, the score of the joint probability of the texture features and RCFs, $\tilde{Rj}(T,RCF)$, can be modeled as for a single feature:

$$Rj(T,RCF) = \log \frac{P(T,RCF|O)}{P(T,RCF|\overline{O})} \qquad \tilde{Rj}(T,RCF) = \frac{P(T,RCF|O)}{P(T,RCF|O)+P(T,RCF|\overline{O})}$$



(a) (b) (c) (d) (e) (f)

Figure 4: Examples of results on the Spotted Cats and Cars datasets. Columns (a) and (d) show the original images, (b) and (e) shown the results of using single region classification with feature TR(1000,300) for the Spotted Cats and RCFs for the Cars, and (c) and (f) show the results of incorporating neighbourhood information. Results are in the 'jet' colourmap, ranging from dark blue for background to dark red for object detections.

In Figure 3 we compare the effects of using the two feature combination models. For clarity, only two representative curves per model are plotted. The combination of RCFs with the TR texture representation with $(K_{TEX}, K_{TR}) = (1000, 50)$ provided the best results for both combination methods, surpassing even $(K_{TEX}, K_{TR}) = (1000, 300)$. Despite a large amount of training data, the best results are not always obtained with the largest dictionaries. Another counter-intuitive result is that combining the features with an assumption of independence performs better than modeling the joint distribution even on the Cars dataset which contains more than 1000 training images. Finally, note that although the combination $(K_{TEX}, K_{TR}) = (200, 300)$ performs very poorly in the joint distribution representation, it outperforms $(K_{TEX}, K_{TR}) = (1000, 50)$ for high recall values in the independent distribution representation. This indicates that performance under one type of distribution does not mirror performance under the other. This result can be applied to other object recognition systems, suggesting that many common assumptions in system implementations should in fact be verified.

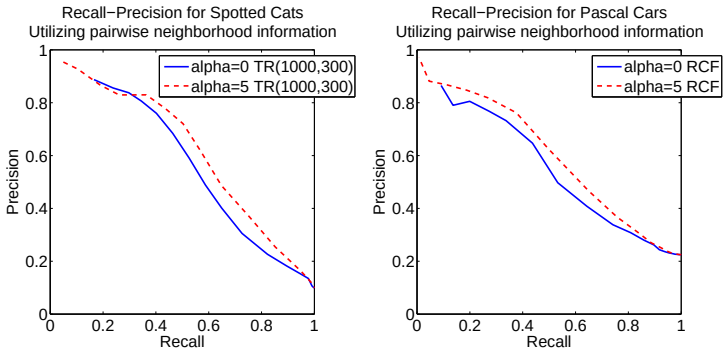
5 Incorporating neighbourhood information

We have made progress in classifying regions to detect and segment objects, however we still make mistakes, especially with respect to spurious object detections and missed detections which are inconsistent with the regions around them, such as in columns (b) and (e) of Figure 4. In this section we describe a method for smoothing the localization results using information from adjacent pairs of regions which leads to the results in columns (c) and (f) of Figure 4.

To incorporate pairwise information, the two-class classification problem can be reformulated as an energy minimization. Let $x_i \in \{0,1\}$ be the label of region i, with 0 for background and 1 for object. Then $\mathbf{X} = [x_1..x_N]$ is the vector of assignments to every region in an image. Also, let F_i be the feature cluster for region i, and $\mathbf{F} = [F_1..F_N]$ be the vector of region features over the entire image. Here F can be a texture feature or an RCF. We define the energy of an assignment of labels X to an image to be:

$$E(\mathbf{X}, \mathbf{F}) = \sum_i U(x_i, F_i) + \alpha \sum_i \sum_j B(x_i, x_j, F_i, F_j) \tag{2}$$

Provided that the pairwise term $B(x_i, x_j, F_i, F_j)$ is associative, we can use graph cuts to



(a) (b)

Figure 5: Pixel-level precision-recall curves for the (a) Spotted Cats and (b) Cars datasets using pairwise neighbourhood information. The best-performing single-region representations were used, TR(1000,300) for the Spotted Cats and RCFs for the Cars. Each curve shows the results using a different α as in Equation 2. For $\alpha = 0$ the binary energy term is ignored, and the results mirror those of single regions in Figure 2. From the $\alpha = 5$ curve we see that incorporating neighbourhood information can indeed improve precision by 5-10% for many of the recall values.

minimize the energy exactly and obtain a region labeling [13].

In the energy function, the unary term serves to classify single regions. By using a generative model, we can define the unary energy term as:

$$-\log P(F_i|x_i) = -\log P(F_i|x_i{=}0)^{(1-x_i)} P(F_i|x_i{=}1)^{x_i} = -x_i \log \frac{P(F_i|x_i{=}1)}{P(F_i|x_i{=}0)} - \log P(F_i|x_i{=}0)$$

Note that $\log P(F_i|x_i = 0)$ is constant with respect to the label assigned to x_i, so we can ignore it in the energy minimization. Also, as in the single-region classifier, we can use the bounded posterior $\tilde{R}(F_i)$ instead of the unbounded log likelihood ratio. Note that this is an approximation computed using *image* labels, not *region* labels. Finally, energy minimization via graph cuts produces a binary labeling of the data, but it does not produce marginals. To compute different recall-precision values, we use a penalty term β which adjusts the threshold between declaring a region part of an object or the background. Another method for achieving this goal is presented in [12]. Our final unary term is:

$$U(x_i, F_i) = -x_i \left(\tilde{R}(F_i) - \beta\right)$$

In most energy minimizations, the binary energy term is a classifier between the four possible assignments $(x_i, x_j) \in \{(0,0),(0,1),(1,0),(1,1)\}$ trained using fully labeled data. Since we only have image-level labels, defining and training the binary energy term is a more difficult process. We choose to define $B(x_i, x_j, F_i, F_j)$ as a smoothing term which only differentiates between the two regions having the same label or different labels. The lack of training data with adjacent neighbours labeled $(0,1)$ or $(1,0)$ makes this impossible to train directly, so we define the binary terms as:

$$B(x_i, x_j, F_i, F_j) \quad = \quad \begin{cases} log(1 + \frac{P(F_i, F_j | i \rtimes j)}{P(F_i, F* | i \rtimes j) + P(F*, F_j | i \rtimes j)}) & x_i \neq x_j \\ 0 & x_i = x_j \end{cases} \tag{3}$$

Where $i \bowtie j$ means that i and j are neighbours, and $F*$ is any feature cluster. The $1+$ term ensures that $B >= 0$ and hence is associative. By this definition of B, we prefer to smooth region labels if the regions are frequently seen together and hence are likely to belong to the same object. Features which are rarely seen together are more likely to belong to different objects, hence their respective region labels are less prone to smoothing.

We performed experiments using this energy function on both datasets. The features used were the TR(1000,300) for the Spotted Cats, and the RCFs for the Cars, since they performed best on the single-region representation task. Values of α of 0, 0.5, 1 and 5 were used. In Figure 5 we show the results of setting $\alpha = 0$ and $\alpha = 5$. We omit the $\alpha = 0.5$ and $\alpha = 1$ settings for clarity since they lie strictly between the $\alpha = 0$ and $\alpha = 5$ curves. If $\alpha = 0$, the binary term in the energy function is ignored, and indeed this curve matches the single-representation RCF curve in Figure 2.

From the $\alpha = 5$ curve we can see that using this pair-wise smoothing approach does indeed improve precision by 5-10% for most recall values. Example results are given in Figure 4.

6 Conclusions

In this paper we have presented a study of issues related to using segmentation-generated regions for object class recognition and segmentation in a weakly supervised learning framework through a particular instantiation of such a system. Results were computed for pixel-level localization. We explored three different region representations and model selection issues for each, concluding that the choices are dataset-dependent and hence multiple representations should be evaluated for future systems in this category.

In order to reduce the burden of region representation, we examined ways to combine representations that would perform at least as well as a single representation alone. Our results showed that modeling the joint distribution of the representations by assuming their independence can lead to promising performance. However, even a large dataset such as the 1000 training images we used for the PASCAL Cars was not enough to properly model a full joint distribution between representations.

Finally, we attempted to reduce the spurious object detections and spurious object misses by incorporating information from a region's neighbourhood. Both single regions and pairs of neighbouring regions were used to compute an energy-minimizing labeling solution which outperformed the use of single regions alone.

References

[1] K. Barnard, P. Duygulu, R. Guru, P. Gabbur, and D. Forsyth. The effects of segmentation and feature choice in a translation model of object recognition. In *CVPR*, 2003.

[2] N. W. Campbell, B. T. Thomas, and T. Troscianko. A two-stage process for accurate image segmentation. In *Int. Conf. on Image Processing and its Applications*, 1997.

[3] P. Carbonetto, N. de Freitas, and K. Barnard. A statistical model for general contextual object recognition. In *ECCV*, 2004.

[4] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *PAMI*, 2002.

[5] G. Dorkó and C. Schmid. Selection of scale-invariant parts for object class recognition. In *ICCV*, volume 1, pages 634–640, 2003.

[6] P. Duygulu, K. Barnard, N. de Freitas, and D. A. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *ECCV*, 2002.

[7] M. Everingham, L. V. Gool, C. Williams, and A. Zisserman. The pascal 2006 visual object classes challenge.

[8] M. R. Everingham, B. T. Thomas, and T. Troscianko. Head-mounted mobility aid for low vision using scene classification techniques. *Int'l Journal of Virtual Reality*, 3, 1999.

[9] X. He, R. Zemel, and D. Ray. Learning and incorporating top-down cues in image segmentation. In *ECCV*, 2006.

[10] D. Hoiem, A. A. Efros, and M. Hebert. Geometric context from a single image. In *Proceedings of ICCV*, 2005.

[11] I. Y. Kim and H. S. Yang. Efficient image understanding based on the markov random field model and error backpropagation network. *PAMI*, 1992.

[12] P. Kohli and P. Torr. Measuring uncertainty in graph cut solutions - efficiently computing min-marginal energies using dynamic graph cuts. In *ECCV*, 2006.

[13] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? In *ECCV*, 2002.

[14] M. Kumar, P. Torr, and A. Zisserman. Obj cut. In *CVPR*, 2005.

[15] B. Leibe and B. Schiele. Interleaved object categorization and segmentation. In *BMVC*, 2003.

[16] A. Levin and Y. Weiss. Learning to combine bottom-up and top-down segmentation. In *ECCV*, 2006.

[17] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[18] W. P. J. Mackeown, P. Greenway, B. T. Thomas, and W. A. Wright. Labelling images with a neural network. In *ICANN*, 1993.

[19] J. Malik, S. Belongie, J. Shi, and T. Leung. Textons, contours and regions: Cue combination in image segmentation. In *ICCV*, 1999.

[20] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, 2001.

[21] J. W. Modestino and J. Zhang. A markov random field model-based approach to image interpretation. *PAMI*, 14, 1992.

[22] A. Opelt and A. Pinz. Object localization with boosting and weak supervision for generic object recognition. In *SCIA*, 2005.

[23] C. Pantofaru, G. Dorkó, C. Schmid, and M. Hebert. Combining regions and patches for object class localization. In *Beyond Patches Workshop, CVPR*, 2006.

[24] C. Pantofaru and M. Hebert. A comparison of image segmentation algorithms. Technical Report CMU-RI-TR-05-40, Robotics Institute, Carnegie Mellon University, September 2005.

[25] B. Russell, A. Efros, J. Sivic, W. Freeman, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *Proceedings of CVPR*, June 2006.

[26] S. Savarese, J. Winn, and A. Criminisi. Discriminative object class models of appearance and shape by correlatons. In *CVPR*, 2006.

[27] C. Schmid. Constructing models for content-based image retrieval. In *CVPR*, 2001.

[28] F. Schroff, A. Criminisi, and A. Zisserman. Single-histogram class models for image segmentation. In *ICVGIP*, 2006.

[29] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, 2006.

[30] Z. Tu, Z. Chen, A. L. Yuille, and S.-C. Zhu. Image parsing: Unifying segmentation, detection, and recognition. *IJCV*, 2005.

[31] F. Vivarelli and C. Williams. Using bayesian neural networks to classify segmented images. Technical report, Aston University, 1997.

[32] J. Winn and J. Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *CVPR*, 2006.

Image Enhancement Using Vector Quantisation Based Interpolation

W. Paul Cockshott, Sumitha L. Balasuriya, Irwan Prasetya Gunawan, and J. Paul Siebert

University of Glasgow, Computing Science Department

17 Lilybank Gardens, Glasgow G12 8QQ

{wpc,sumitha,ipguna,psiebert}@dcs.gla.ac.uk

Abstract

We present a novel method of image expansion using vector quantisation. The algorithm is inspired by fractal coding and uses a statistical model of the relationship between details at different scales of the image to interpolate detail at one octave above the highest spatial frequency in the original image. Our method aims at overcoming the drawbacks associated with traditional approaches such as pixel interpolation, which smoothes the scaled-up images, or fractal coding, which bears high computational cost and has limited use due to patent restrictions. The proposed method is able to regenerate plausible image detail that was irretrievable when traditional approaches are used. The vector quantisation-based method outperforms conventional approaches in terms of both objective and subjective evaluations.

1 Introduction

Digital cinema sequences can be captured at a number of different resolutions, for example 2K pixels across or 4K pixels across. The cameras used for high resolutions are expensive and the data files they produce are large. Because of this, studios may chose to capture some sequences at lower resolution and others at high resolution. The different resolution sequences are later merged during post production. The merger requires that some form of image expansion be performed on the lower resolution sequences. In this paper we present a new method of doing the image expansion that has some advantages over the orthodox interpolation methods.

The paper is organised as follows. Section 2 will review some of the existing techniques of image expansion and highlight their shortcomings. In Section 3, we will describe the proposed algorithm in details including the process of training the algorithm, constructing the library used in it, and producing as well as enhancing the expanded image using the algorithm. Section 4 contains our experimental results in which our proposed method is evaluated. The paper concludes in Section 5.

2 Background

In the traditional pixel interpolation method, new pixels are generated in the scaled-up image; however, there is no information pertaining to what these pixels should contain, other than interpolation of the original pixels by some polynomial function. Because the polynomial function works over a neighbourhood in the original, smaller image, the scaled-up image will contain less energy at the highest spatial frequencies than the original, making it looks smoother.

An alternative approach through fractal encoding, originally reported by Barnsley [1], allows rescaled images to contain new high frequency information. Fractal encoding takes advantage of the self similarity across scales of natural scenes. A fractal code for an image consists of a set of contractive affine maps from the image, onto the image. Taken as a whole, these maps compose a collage such that each pixel is mapped onto by at least one such map. The maps operate both in the spatial and the luminance domain. In the luminance domain they specify a target pixel p by an equation of the form $p = a + bq$ where q is the mean brightness of a downsample region of source pixels. In the spatial domain they specify the coordinates of the source pixels supporting q as the result of rotation, scaling and translation operations on the coordinates of the destination pixels.

The image is regenerated from the codes by iterated application of the affine maps. The iteration process has an attractor that is the output image. If the maps have been well chosen this attractor approximates well to a chosen input image.

A particular fractal code might specify each 4x4 rectangle within a 256x256 pixel output image in terms of a contractive map on some 8x8 rectangle at some other point in the image. As the iteration proceeds higher and higher frequency information is built up. If we start from a uniform grey image, the first iteration will generate detail at a spatial frequency of 8 pixels. After one iteration source blocks of 8 pixels across will contain up to one spatial wave. After the second interpolation these waves will have been shifted up in frequency to 4 pixels across. Each iteration adds detail one octave higher until the Nyquist limit of the output image is reached: 128 spatial cycles in this case.

It is evident that if we specify the contractive mappings relative to the scale of the whole image rather than in terms of pixels, then the same set of mappings could be used to generate a 512x512 pixel output image. In this case the contractive mappings would shrink 16 pixel blocks to 8 pixel blocks. After an additional round of iteration the 512 pixel output image will contain spatial frequencies up to 256 cycles.

Fractal codes can thus be used to expand an image, generating new and higher spatial frequencies in the process. Although the additional detail that is added by this process can not have been available in the source image it nevertheless 'looks plausible' because the 'new' details are scaled down versions of details that were present in the original picture (see Figure 1). The search process used in a fractal encoder scans a half sized copy of the original image to find a match for each small block in the original image. In fractal enhancement the small blocks are then replaced by their full sized corresponding blocks. The detail enhancement comes because there is a *systematic relationship* between the low frequency and high frequency information within blocks. This allows high frequency information in a larger block to be plausibly substituted into a smaller block when the latter is enlarged.

Fractal codes however suffer from two serious obstacles to their widespread adoption: the encoding algorithm is slow and their general use is blocked by patent restrictions. In

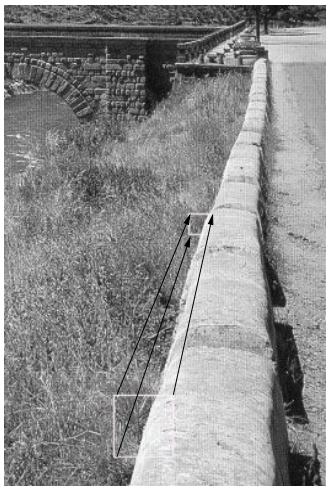


Figure 1: Illustration of how shrinking is used to fill in detail in fractal enhancement.

this paper we present an alternative approach that learns lessons from fractal coding but avoids these difficulties. Instead of using fractals we use vector quantisation to enhance the detail of an image.

3 Proposed method

The key idea of our approach is that because there is a systematic relationship between low and high frequency information within a neighbourhood, it should be possible for a machine learning algorithm to discover what this relationship is and exploit this knowledge when enhancing an image. We use vector quantisation to categorise areas of the image at different scales, learn the systematic relationship between the coding of corresponding areas at varying scales, and then use this information to extrapolate a more detailed image. The entire process works by

1. Running a training algorithm to learn the cross-scale structure relations in example pictures. In the experiments here two images were used; one from the 'face' sequence and one from the outside 'trees' sequence.
2. Using this information to automatically construct a new image enhancing program.
3. Applying the enhancing program to digital cine images to generate new images at twice the resolution.

3.1 The Training Algorithm

The aim of the training algorithm is to learn what high frequency detail is likely to be associated with the low frequency features at a given point in an image. Given an image I we construct a half sized version of the image $I_{0.5}$ and expand this to form a new blurred image I_b which is the original size, by using linear interpolation. We now form a difference image $I_d = I - I_b$ which contains only the high frequency details.

It is clear that we have a genetive association between position $I_{0.5}[x,y]$ and the four pixel block $G_{x,y} = \{I_d[2x,2y], I_d[2x+1,2y], I_d[2x,2y+1], I_d[2x+1,2y+1]\}$. We aim to categorise the regions around each position in $I_{0.5}[x,y]$, categorise the corresponding blocks $G_{x,y}$ and learn the associations between these categories.

3.1.1 Categorising the Upper Layer

Associate with each pixel $p \in I_{0.5}$ a neighbourhood and compute the differences between p and its neighbours. These define a 4 element vector. Using the algorithm given in [6] construct a vector quantisation codebook B_1 for these features. Assume that the code book has n entries.

3.1.2 Categorising the Lower Layer

Use the same vector quantisation algorithm to construct a second vector quantisation codebook B_2 for the set of vectors $G_{x,y}$. Assume that the code book again has n entries.

3.1.3 Learning the Association

Encode the neighbourhoods around each pixel $p \in I_{0.5}$ with B_1 to yield an encoded image $E_{0.5}$. Encode each $G_{x,y}$ associated with each pixel $p \in I_{0.5}$ with B_2 to yield an encoded image E_b. The entries in both the encoded images are indexed into the respective codebooks.

Construct an $n \times n$ frequency table F that counts how frequently each code from B_1 is associated with each code from B_2. Finally convert the frequency table to a conditional probability table by dividing by the number of observations.

3.2 The Program Generator

The aim of the program generator is to take the tables B_1, B_2, F and use them to generate pascal libraries that can be used to index and predict detail in subsequent images. The process is analogous to the way Lex [5] constructs scanner tables in C from a regular grammar.

Two optimisations are performed prior to outputting the tables:

1. Table F is converted from a conditional probability table to a table encoding the cumulative probability of each entry in B_2 being associated with and entry in B_1.
2. Hierarchical Vector Quantisation [2] indices are constructed for the two codebooks to enable future encoding to be of $\mathbf{O}(4)$ rather than $\mathbf{O}(\mathrm{n})$.

3.3 The Enhancement Algorithm

The enhancement program has the library produced above linked to it. The aim of the program is to read in an image J and produce an image J_2 of twice the size with enhanced detail. The algorithm is described by Algorithm 1.

Once this process has been completed for each pixel in J the image J_d contains details whose spatial frequency is one octave higher than those that are represented in J_b. Each

Algorithm 1 Proposed method

1: Create image J_b of twice the size of J using linear interpolation.
2: Create an empty image J_d twice the size of J.
3: **for all** pixel $p \in J$ at position x, y **do**
4: Compute its differences with its four neighbours as described in 3.1.1
5: Encode the four differences using a vector quantisation index for book B_1 to yield a code index number i
6: Select the ith row of F
7: Draw a real number r at random such that $0 \leq r < 1$
8: Scan row $F[i]$ until $F[i, j] > r$ is found
9: Select the 4 element vector $B_2[j]$.
10: Place this vector in the image J_d at positions $\{J_d[2x,2y], J_d[2x+1,2y], J_d[2x,2y+1], J_d[2x+1,2y+1]\}$.
11: **end for**
12: Forms the enhanced image J_2 by the operation $J_2 = J_b + J_d$

detail occurs with the same probability with respect to the categorisation of localities in J as details occurred in I_d with respect to the categorisation of localities in $I_{0.5}$.

4 Results

Our experiments were conducted on 1920×1080 video frames in the DPX image format captured by a Thompson Viper D-Cinema video camera. The pixels were in 10 bit logarithmic format. Image expansion using our system was compared to conventional bilinear and B-spline interpolation techniques. The experimental procedure can be described as follows:

1. The enhancement system was trained on a 1920×1080 DPX frame from an outdoor sequence. The test images used were a studio frame (Figure 2), a later frame from the outdoor sequence from which original training frame had been selected (Figure 4), and a train sequence (Figure 6). The training and test frame from this sequence had different zoom settings, the training frame having had a higher zoom factor than the second test frame.
2. The test images were downsampled using bilinear interpolation to 960×540 and output in DPX format.
3. Image expansion to double the original resolution of 1920×1080 was performed using our algorithm, bilinear and B-spline interpolations.
4. Original and expanded DPX video frames were compared subjectively based on perceived detail in image patches. The quality of reproduction was also evaluated objectively using several image quality metrics described below.

We used several methods of assessment to evaluate the quality of the reconstructed images. First, a traditional measure based on Peak Signal-to-Noise Ratio (PSNR) [7] was calculated. In this paper, the PSNR was calculated on the 10-bit logarithmic representation of pixel values. This metric is very practical and easy to compute, however critic says that it does not always correlate well with the quality perceived by human users [3]. An alternative using a modified version of the PSNR based on perceived visibility of error, namely Weighted Peak Signal-to-Noise Ratio (WPSNR) [8], was also computed. In this metric, error on textured area would be given less weighting factor than that on flat surface.

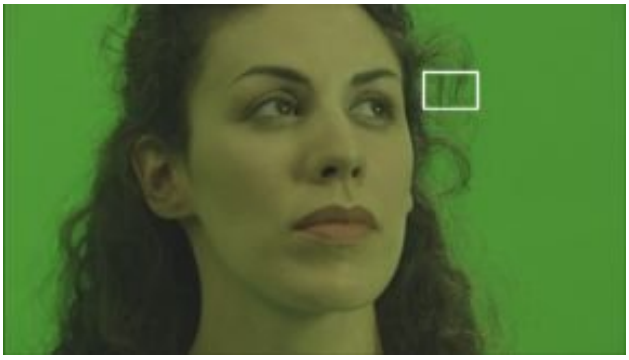
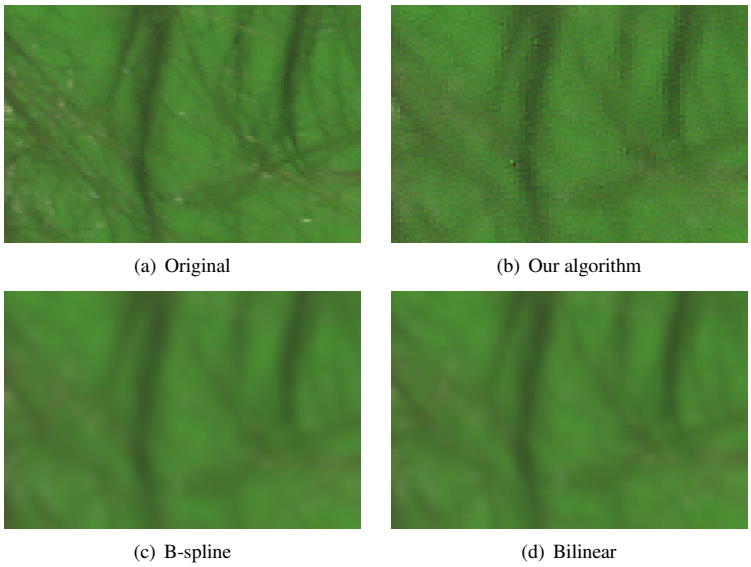


Figure 2: Reduced scale colour image from original DPX digital cine frame from studio sequence ‘face’. Relatively soft focus is used with a moving subject. Box indicates where detail is shown in Figure 3. Note that this and all following images are uncorrected log colour space.



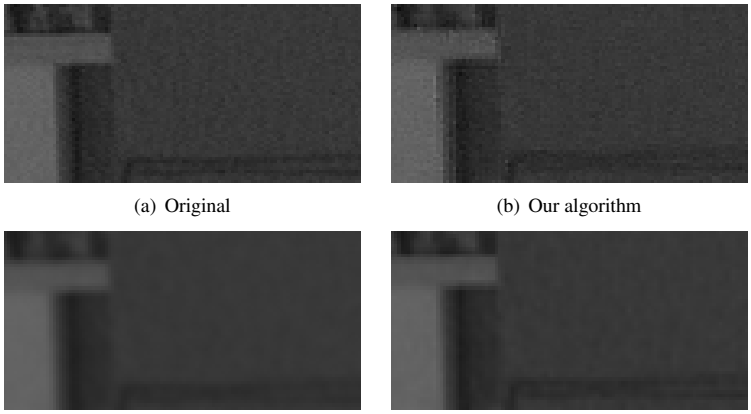
(a) Original (b) Our algorithm

(c) B-spline (d) Bilinear

Figure 3: A region with high frequency detail from original DPX video frame from the ’face’ sequence (a) and the corresponding region generated by expansion of the half-resolution video frame by our algorithm (b), as well as by B-spline interpolation (c) and bilinear interpolation (d).

Since image expansion algorithms usually introduce blur artifacts, a quality metric [4] capable of detecting and measuring the degree of blurriness on the degraded image was used. This metric computes features extracted from the frequency domain through two-dimensional Discrete Fourier Transfrom (DFT) computation over a localised area on the gradient image. In an image contaminated by blurring distortion, some frequency components appear attenuated when compared to the corresponding set of components on the original image. Blurriness detection can be done by analysing the decay in the

Figure 4: Image taken outside in bright light, with sharp focus containing more high frequency detail. Box shows area used in Figure 5.



(a) Original (b) Our algorithm

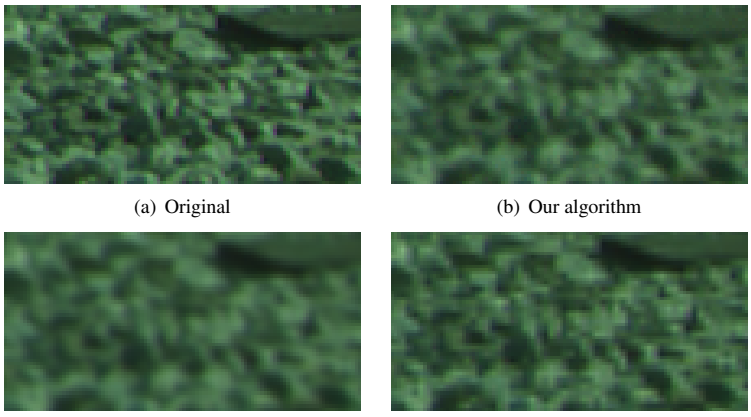
(c) B-spline (d) Bilinear

Figure 5: Samples taken from the frame shown in Figure 4. The algorithm convincingly synthesises speckle on the concrete wall but leaves the white wall in the background speckle free.

strength of these frequency components. One of the quality parameters produced by this metric is harmonic loss blur index, which we used to measure the degree of blurriness on our images in this paper. Because blur index is a distortion metric, higher value of the index on an image implies that the image contains more blurring artifacts.

In addition to the above metrics, we also use another quality metric that is sensitive to structural distortions. Blurriness in the expanded images may appear as a result of edge widening as well as change of contrast. The universal quality index (UQI) [9], which is a special case of structural similarity (SSIM) index, has the ability to measure such structural distortions due to image expansion. This metric is significantly different from the traditional PSNR/WPSNR since the latter attempt to quantify the visibility of errors between the degraded images and the original image, whilst UQI models any distortions as a combination of three different factors: loss of correlation, luminance distortion, and

Figure 6: Image from 'train' sequence. The details are shown in Figure 7.



(a) Original (b) Our algorithm

(c) B-spline (d) Bilinear

Figure 7: Detail samples from 'train' sequence in Figure 6.

contrast distortion. This metric has a dynamic range of $[-1, 1]$, with 1 is the best value (correspond to highest quality).

It is subjectively apparent that our algorithm has regenerated plausible image detail that was irretrievable when using the B-spline and bilinear interpolation approaches (Figures 3, 5, and 7). The down-sampling suppressed visual information which only our algorithm could recover based on its knowledge of statistical co-occurrence of low and high frequency image content. It was also observed that VQ-based enhancement method was better than conventional method (e.g. B-spline) since the latter introduces more blurriness to the processed images. The above mentioned figures show the degree of the blurriness of details from these sequences which have been enhanced by three different methods (our proposed VQ-based, bilinear, and B-spline).

Objective comparison of our algorithm with bilinear and B-spline interpolations for image expansion are shown in Table 1. As an example, Figure 7, which was taken from the 'train' sequence and excluded during the training stage, is used for the computation of

Objective quality metrics	Proposed method	Bilinear	B-Spline
PSNR [7]	36.9 dB	28.4 dB	36.6 dB
WPSNR [8]	44.8 dB	39.2 dB	44.5 dB
Blur index [4]	0.180	0.193	0.298
UQI [9]	0.836	0.456	0.826

Table 1: Objective image quality metrics for the image expansions shown in Figure 7.

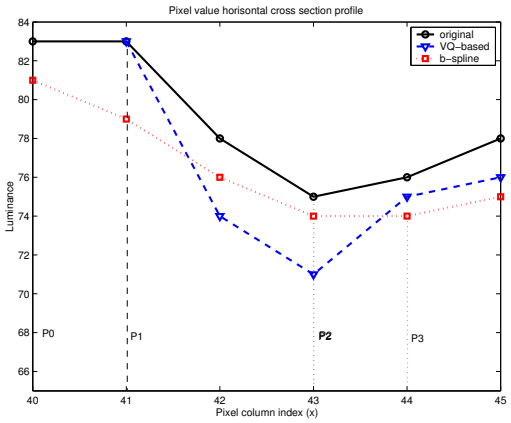


Figure 8: Cross-sectional luminosity profile

these metrics. Our method is superior than the others in terms of the objective measures used in this paper. For example, the blur index values are in favour of our VQ-based method; it shows that the blurriness indices of the B-spline and bilinear enhanced images are generally higher than those of VQ-based, with B-spline enhanced image is the worst in quality (i.e., contains the most blur distortions). The bilinear method, although comparable with our method in terms of blur index, has the most damaged structure as shown by the UQI values.

The blurring effect of expansion method on edges can also be illustrated by observing pixel-value cross-section along a row (or column) in an image. Cross-sectional luminosity profile of the detail images (Figure 7) are shown in Figure 8. In this figure, the local maxima around edges are indicated by the dashed lines ($P0$ and $P1$) and local minima by the dotted lines ($P2$ and $P3$). The graph shows that the edge in the image expanded by the B-spline method is wider (with an edge width of $P3 - P0$) than those in the original and the VQ-based enhanced images (edge width of $P2 - P1$). It explains that the blur in the traditional expansion method is due to the increased edge width.

5 Conclusions

This paper presented a novel approach to image enhancement using a technique which would avoid the known shortcomings of fractal enhancement. We learnt the statistical properties of the co-occurrence of low and high frequency image content and used these

probability distributions to predict image content during image expansion. Subjective evaluations show that our algorithm outperforms conventional approaches. This is also confirmed by the results of objective evaluations based on several published quality metrics suitable for the application presented here. We hope to continue working on our algorithm which is still in its preliminary stages by: 1) Learning statistical co-occurrence of neighbouring codebook blocks in images; and 2) Mediating the addition of high frequency predicted detail with the energy of the underlying region in the image to prevent prediction of detail in the absence of high frequency information in the original image.

Acknowledgements

This work is supported by the European Commission under the IP-RACINE project (IST-2-511316-IP).

References

[1] M.F. Barnsley and L.P. Hurd. *Fractal image compression*. AK Peters, 1993.

[2] N. Chaddha, M. Vishwanath, and PA Chou. Hierarchical vector quantization of perceptually weighted block transforms. *Proceedings of the Conference on Data Compression*, 1995.

[3] Bernd Girod. What's wrong with mean-squared error? In A. B. Watson, editor, *Digital Images and Human Vision*, pages 207–220. 1993.

[4] Irwan Prasetya Gunawan and Mohammed Ghanbari. Image quality assessment based on harmonics gain/loss information. In *Proceedings of ICIP'05 (IEEE International Conference on Image Processing)*, volume 1, pages 429–432, Genoa, Italia, September 11–14 2005.

[5] J.R. Levine, T. Mason, and D. Brown. *lex & yacc*. O'Reilly & Associates, Inc. Sebastopol, CA, USA, 1992.

[6] Y. Linde, A. Buzo, and R. Gray. An Algorithm for Vector Quantizer Design. *IEEE Transactions on Communications*, 28(1):84–95, 1980.

[7] William K. Pratt. *Digital Image Processing*. John Wiley and Sons, 1978.

[8] Sviatoslav Voloshynovskiy, Alexander Herrigel, Nazanin Baumgärtner, and Thierry Pun. A stochastic approach to content adaptive digital image watermarking. In *International Workshop on Information Hiding*, volume LNCS 1768 of *Lecture Notes in Computer Science*, pages 212–236, Dresden, Germany, 29 September –1 October 1999. Springer Verlag.

[9] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transaction on Image Processing*, 13(4):600–612, April 2004.

Camera calibration for miniature, low-cost, wide-angle imaging systems *

Oliver Frank, Roman Katz, Christel-Loic Tisse and Hugh Durrant-Whyte
ARC Centre of Excellence for Autonomous Systems
University of Sydney, Australia
{o.frank,r.katz,c.tisse,h.durrant-whyte}@cas.edu.au

Abstract

This paper presents a new model and an extension to an existing algorithm for camera calibration. The main goal of the proposed approach is to calibrate miniature, low-cost, wide-angle fisheye lenses. The model has been verified with a calibration implementation and was tested on real data. Experiments show that the proposed model improves the accuracy compared to the original algorithm. Results show that the extension not only performs well with fisheye lenses but also with omnidirectional catadioptric lenses as well as other less distorted dioptric lenses.

1 Motivation and Related Work

Wide-angle, hemispherical or omnidirectional camera systems have become more popular in the last few years. Especially in robotic applications, wide-angle sensors are favourable for perception and navigation problems. A precise calibration is needed in order to infer accurate bearing information for the 2D pixel information.

Three different types of calibration methods can be distinguished. The most common approaches are based on the correspondence of feature points whose world 3D coordinates are known [14, 10, 4, 6]. These methods find the best external and internal parameters that correspond to the position of the feature points in the image. A second group of methods [1, 13] uses geometric invariants instead of world coordinates of image features (plumb lines, image of a sphere). The third group, the auto-calibration methods, does not need any kind of known feature points nor geometric invariants. Instead they only use constraints on the internal and external parameters. Their problem can be stability, as ambiguous motion sequences can lead to calibration degeneracies [3, 11].

Hemispherical or omnidirectional cameras imply additional challenges to the calibration procedure that go beyond affine or perspective projection. The single viewpoint constraint does not necessarily hold anymore and lens distortion plays a crucial role. To overcome the multiple viewpoint problem, non-central camera models were introduced. They project scenes onto images along a general set of rays that do not meet in a single point [7], [12]. An intermediate class of cameras are the axial cameras in which the projection rays intersect on a line [9].

*This work is supported by the ARC Centre of Excellence programme, funded by the Australian Research Council (ARC) and the New South Wales state government.

The problem of image distortion caused by lens abberation or misalignment of the lenses has been studied extensively. Not only is the radial distortion of importance but also the decentering (aka tangential) distortion. The plumb lines were introduced in [2] which investigates the relation between straight lines in 3D and their distorted curved projection on the image sensor caused by lens distortion. Most of these models assume the centre of distortion to be at the principal point.

The proposed calibration tool presents an extension to the algorithm presented in [10] and is also motivated by the work of Bouguet[1]. It relies on the use of a planar calibration pattern. The only user input required is the selection of feature points with known 3D coordinates. More recently, an approach based on exact theoretical projection functions has been published in [6]. In this approach, deviations from the exact projection functions are modelled with added error parameters.

This paper is organised as follows. Section 2 introduces the proposed camera model. The process of estimating the calibration parameters is briefly explained in 3 and can be found in more detail in [10]. Section 4 shows experimental results with real data and compares the proposed calibration procedure with the original approach as well as with the algorithm presented in [6]. Finally, conclusions are drawn in Section 5.

2 Camera Model

The proposed model is based on the following assumptions: (i) the lens system has a single effective viewpoint, (ii) there is a misalignment of the image sensor with respect to the optical axis, (iii) non-radial distortion is ignored and the lens system is rotationally symmetrical with respect to its axis.

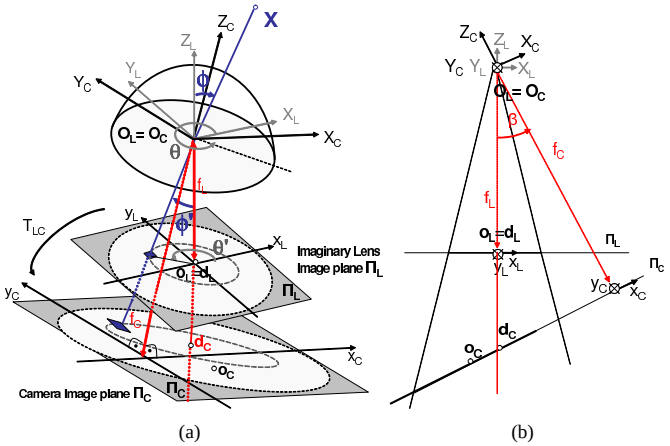
The first assumption of a single effective viewpoint is used for simplicity. Especially for miniature fisheye lenses with very short focal length this is a reasonable approximation. The second assumption is particularly motivated by low cost imaging systems. Misalignment can be caused by the inaccurate mounting of the lens as well as by misplacement of the image sensor relative to the optical axis. Contrary to the model presented in [10], the misalignment is modelled based on a perspective transformation instead of an affinity. Four degrees of freedom (DOF) are sufficient, as thoroughly explained in Section 2.3. The last assumption can be justified by the accurate manufacturing of lenses and the observation that non-radial distortion is mainly caused by the misalignment of different lens components.

The proposed camera model consists of the centre of distortion, the lens distortion function and the internal position of the sensor with respect to the lens. It is a good compromise between fully general camera models and conventional models in terms of complexity and stability of the calibration process.

2.1 Geometry

The proposed model can be described in two stages considering two different coordinate systems. The first one being the lens coordinate system (X_L, Y_L, Z_L) which relates the orientation and position of the lens with respect to the world coordinates of the scene points

[1] Jean-Yves Bouguet. Camera Calibration Toolbox for Matlab, 2007
`http://www.vision.caltech.edu/bouguetj/calib_doc/index.html`



(a) (b)

Figure 1: (a) Imaging model consisting of a radially symmetric distortion lens, an imaginary, perfectly aligned (principal point on optical axis and image plane perpendicular to the optical axis) lens image plane Π_L and the misaligned physical image plane Π_C. (b) Top view for an example misalignment with rotation about Y_L and change of focal length.

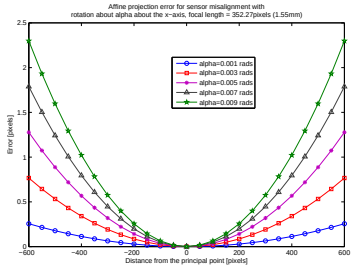
X. The second coordinate system (X_C, Y_C, Z_C) represents the orientation and position of the camera image sensor with respect to the lens coordinate system.

As illustrated in Figure 1, the incident 3D rays from scene points enter the lens system and, after a radial distortion, they are projected onto the imaginary lens image plane Π_L. Π_L is the perpendicular plane to the optical axis Z_L, and the principal point $\mathbf{o_L}$ is the intersection point of the optical axis Z_L and the image plane Π_L. The distorted rays originating from the centre of projection $\mathbf{O_C}$ do not physically intersect with the lens image plane Π_L but with the camera image plane Π_C instead. The physically captured image $\mathbf{x_C}$ can be thought of a 3D perspective projection of the imaginary lens image $\mathbf{x_L}$ with centre of projection $\mathbf{O_C}$ equal to $\mathbf{O_L}$. Therefore, the 3D transformation T_{LC} from Π_L to Π_C can be described with a rotation and scale factor. Thus, only four parameters are sufficient to model the misalignment. In summary, the system can be described as: (1) Perfectly aligned projection with radially symmetric distortion, (2) Perspective projection of an imaginary planar scene Π_L onto the camera image plane Π_C with fixed centre of projection.

For fisheye lenses a perspective projection is more appropriate, contrary to other approaches, including the original algorithm presented in [10], where an affine projection is used to model misalignment. As explained in [5], the error between the true perspective image point and its affine approximation is given by

$$\mathbf{x}_{\text{aff}} - \mathbf{x}_{\text{proj}} = \frac{\Delta}{d_0}(\mathbf{x}_{\text{proj}} - \mathbf{x_0}), \tag{1}$$

with Δ denoting the depth relief, d_0 the average depth and $\mathbf{x}_{\text{proj}} - \mathbf{x_0}$ the distance of the point from the principal ray. For a misalignment between lens and sensor, the error is therefore inversely proportional to the focal length. Figure 2 shows the error of the affine



(a)

Figure 2: Error of affine projection versus distance from principal point. The errors are plotted for different misalignments due to a rotation α about the lens-axis Y_L.

projection for a variation of misalignments due to a rotation α about the lens-axis Y_L. The parameters were taken from a typical miniature, wide-angle imaging system as the one presented in Section 4. It can be seen that even for very small misalignments, the projection error is not negligible anymore.

2.2 Perfectly aligned radially symmetric distortion

For the first step of the calibration it is assumed that the distortion caused by the wide-angle lens is radially symmetric and that the lens image is perfectly aligned with the optical axis of the lens. Rather than representing distortion as an image displacement it is modelled as a varying focal length, depending on the distance to the distortion center.

Let the azimuth angle of an incident ray from an object in the scene be θ and the polar angle ϕ. Let $\mathbf{X}$ denote a 3D ray of a scene point in world coordinates and $\mathbf{x_L}$ the corresponding distorted ray. Without further assumptions, the back projection of $\mathbf{x_L}$ to $\mathbf{X}$ holds the relationship $\mathbf{x_L} = \lambda \mathbf{X}$:

$$\mathbf{x_L} = \begin{bmatrix} x_L \\ y_L \\ g(\rho) \end{bmatrix} \simeq \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathbf{X}, \tag{2}$$

where $g(\rho)$ denotes the radial distortion function and $\rho = \sqrt{x_L^2 + y_L^2}$ is the Euclidean distance from the centre of distortion $\mathbf{o_L}$. The two rays $\mathbf{x_L}$ and $\mathbf{X}$ are generally not equal but have the same direction. Therefore

$$\tan(\phi) = \frac{\sqrt{X^2+Y^2}}{Z} = \frac{\sqrt{x_L^2+y_L^2}}{g(\rho)} = \frac{\rho}{g(\rho)}. \tag{3}$$

As in [10], the distortion function $g(\rho)$ can be defined as

$$g(\rho) = s_0 + s_2\rho^2 + \ldots + s_N\rho^N, \tag{4}$$

which satisfies $\partial g/\partial \rho|_{\rho=0} = 0$. By using (3) and (4)

$$s_0 - \tan(\pi/2 - \phi)\rho + s_2\rho^2 + \ldots + s_N\rho^N = 0, \tag{5}$$

and therefore ρ is found as the root of the above polynomial.

Without loss of generality, the focal length f_L from the lens image to the centre of projection $\mathbf{O_L}$ can be fixed to any value without affecting the calibration. For the case of no distortion, the model has to be coherent with the conventional perspective projection with a pinhole camera. Hence, the relation $g(\rho) = f_L$ can be found and by using (4), the focal length results in $f_L = s_0$. With this particular focal length, the relation between the incident angle ϕ of the 3D ray and the angle of the distorted ray ϕ' can be explained by using (3):

$$\tan(\phi') = \frac{\rho}{s_0}, \tag{6}$$

where ρ can be found as the root of (5) and therefore only depends on the incident angle ϕ of the 3D ray.

The distorted image points on the lens image plane can be found using $\theta' = \theta - \pi$, and equates to $[x_L, y_L]^T = [\rho\cos(\theta'), \rho\sin(\theta')]^T$.

2.3 Perspective projection onto the camera image sensor

The same physical distorted rays $\mathbf{x_L}$ as in Section 2.2 are now being projected onto the camera image sensor and converted in pixel coordinates. This can be thought of a perspective projection from the distorted rays onto the camera image plane.

Without loss of generality the focal length f_L of the imaginary lens image plane can be set to 1 and thus the distorted rays $\mathbf{x_L}$ can be written in homogenous coordinates as $\mathbf{x_{L}}_h = [x_L, y_L, 1, 1]^T$. After a perspective projection with the centre of projection being $O_C = O_L$, the image points $\mathbf{x_C}$ are found in homogenous coordinates as

$$\mathbf{x_{C}}_h = \begin{bmatrix} x_C & y_C & w_C \end{bmatrix}^T = K_C \begin{bmatrix} R_{LC} \mid 0 \end{bmatrix} \mathbf{x_{Lh}} = K_C R_{LC} \mathbf{x_L}, \tag{7}$$

where R_{LC} is the 3×3 rotation matrix from the lens coordinate system to the camera coordinate system. It includes rotations α_C, β_C and γ_C about the axis x_C, y_C and z_C respectively. K_C is the common camera calibration matrix including the focal lengths f_{Cx} and f_{Cy} in pixel coordinates, o_{Cx} and o_{Cy}, which are the coordinates of the principal point, and s_C the skew parameter. In Euclidean coordinates (without s_C, o_{Cx} and o_{Cy}), the projection from the lens coordinate system to the camera coordinate system has four degrees of freedom. It is fully described by a rotation and a scale factor.

3 Calibration Procedure

The calibration procedure presents an extension to the toolbox presented in [10]. It relies on a planar calibration grid which is captured from different poses. The algorithm is based on the popular method introduced by [14].

In order to fully calibrate the camera with the previously introduced model, estimations of all the $8 + N + 6M$ parameters have to be performed. M denotes the number of used calibration images and N is the degree of the highest order term in (4). As found in [10] for the original algorithm and verified for the proposed calibration procedure, best approximations are expected with $N = 4$. For one image i, the parameters are the conventional intrinsic ones s, o_{cx}, o_{cy}, the lens distortion parameters $s_0, s_2, \ldots, s_N$, the

misalignment parameters α_C, β_C, γ_C, f_{Cx}, f_{Cy} and the conventional extrinsic parameters α^i, β^i, γ^i, C_x^i, C_y^i, C_z^i. Note that a conventional perspective model has $5+6M$ parameters.

The calibration procedure can be outlined as follows:

Calibration Procedure:
Given M calibration images, determine the $8+N+6M$ intrinsic parameters.

1. Estimation of the centre of distortion $\mathbf{d}$ for the aligned lens model as the least-squares solution of all images I_i for $i = 1 \ldots M$.
2. Linear estimation of the extrinsic parameters for all the images from $i = 1 \ldots M$ by assuming the aligned model and using the previously estimated centre of distortion $\mathbf{d}$.
3. Linear estimation of the intrinsic parameters with the aligned lens model by using the previously estimated parameters for all images.
4. Optimised refinement and estimation of the remaining parameters with the full misaligned model. Use the linear estimations from 2 and 3 as initialisation values for the optimisation process.

Step 2 and 3 of the calibration procedure are similar to the one presented in [10] and thoroughly explained therein.

3.1 Find the centre of distortion

Step 1, the choice of the distortion centre is crucial for a stable and accurate estimation. As introduced in [4], the problem of radial distortion is similar to the motion of points seen by a camera moving forward towards a scene. The same idea can be used, even if an additional misalignment between the lens and the camera sensor is considered. The ideal, undistorted image $\mathbf{x_u}$ is introduced as the projection of $\mathbf{X}$ with a perfect pinhole camera $\mathbf{x_u} = \mathbf{PX}$. It is related to its radially distorted version $\mathbf{x_L}$ on the imaginary image plane by $\mathbf{x_L} = \mathbf{d_L} + \lambda(\phi) \cdot (\mathbf{x_u} - \mathbf{d_L})$, with $\mathbf{d_L}$ being the distortion centre, and $\lambda(\phi) = \rho/r$ the radially symmetric distortion factor from the distorted to the undistorted Euclidean radius. By writing the relation between the distorted image $\mathbf{x_L}$ and its misaligned version $\mathbf{x_C}$ in (7) as a homography $\mathbf{x_C} = \mathbf{H_{LC}}\mathbf{x_L}$, the following relation can be found:

$$\mathbf{H_{LC}^{-1}}\mathbf{x_C} = \mathbf{d_L} + \lambda(\phi) \cdot (\mathbf{x_u} - \mathbf{d_L}). \tag{8}$$

Multiplying this expression on the left by the skew-symmetric matrix $[\mathbf{d_L}]_x$, and substituting $\mathbf{x_u}$ by $\mathbf{PX}$ equates in

$$[\mathbf{d_L}]_x \mathbf{H_{LC}^{-1}}\mathbf{x_C} = \lambda(\phi)[\mathbf{d_L}]_x \mathbf{PX}. \tag{9}$$

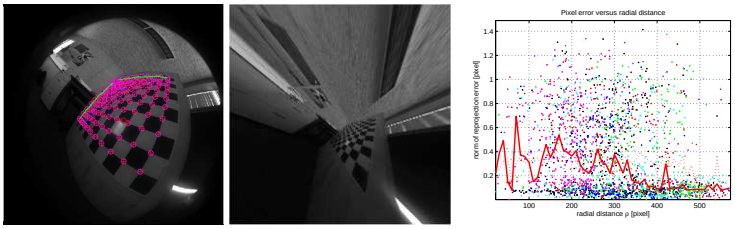
Another multiplication on the left by $(\mathbf{H_{LC}^{-1}}\mathbf{x_C})^T$ results in

$$0 = \lambda(\phi)\mathbf{x_C}^T \left(\mathbf{H_{LC}^{-T}}[\mathbf{d_L}]_x \mathbf{P}\right) \mathbf{X}. \tag{10}$$

This can be compared to the usual fundamental matrix relation $\mathbf{x_L}^T \mathbf{F} \mathbf{x_C} = 0$ with $\mathbf{F}$ being the substitution of $\mathbf{H_{LC}^{-T}}[\mathbf{d_L}]_x \mathbf{P}$. It can be computed from several point correspondences between the calibration pattern $\mathbf{X}$ and the distorted points $\mathbf{x_C}$ in the camera image plane [5]. A final multiplication on the left by $\mathbf{H_{LC}^{T}}\mathbf{d_L}^T$ yields

$$(\mathbf{H_{LC}}\mathbf{d_L})^T \mathbf{F} = 0, \tag{11}$$

and the centre of distortion in the misaligned camera image plane can be found as the left null space generator of F. This can be done with the singular value decomposition of F.



(a) (b) (c)

Figure 3: (a) Sample calibration image with extracted and reprojected grid points overlaid. (b) Image after rectification. (c) Radial distribution of the reprojection errors for all images contained in the set.

3.2 Full misaligned model and non-linear refinement

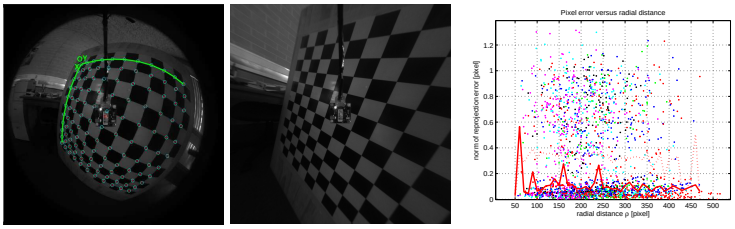
The final step of the calibration process refines the previously estimated intrinsic and extrinsic parameters that were found by assuming a perfectly aligned system. Simultaneously the remaining intrinsic parameters f_{Cx}, f_{Cx}, α_C, β_C and γ_C for the full misaligned camera model are now estimated. This can be done either for the intrinsic and extrinsic parameters independently, or simultaneously with bundle adjustment. The latter is expected to provide more accurate results at higher computational cost. For the experimental results in Section 4 the *Levenberg-Marquardt* [8] method was used to refine all parameters simultaneously.

4 Experimental Results

The proposed camera model and the calibration algorithm were tested on real images from a planar calibration checkerboard. A simple corner detector was used to extract the 'ground truth' grid corners. These values were then compared with the calculated reprojections from the known world coordinates to the camera pixel coordinates. As a performance measurement the mean square reprojection error in pixel coordinates was used. In the following figures the 'ground truth' and reprojected grid corners are represented as ('+') and ('o') respectively.

The first and second data sets, *Fisheye1* and *Fisheye2*, were taken with a *uEye UI-2250* from *Imaging Development Systems* together with a miniature fisheye lens *DSL215* from *Sunex*. The field of view for the miniature fisheye lens is specified at about 185° and the focal length is 1.55mm. The entire data set contains 18 images. Figure 3(a) shows a sample image taken from the imaging system with the extracted corner points (+) and its reprojection (o). Figure 3(b) illustrates the calculated rectified image as if viewed with a perfect pinhole camera system. The radial distribution of the reprojection error for all calibration images in this set is depicted in Figure 3(c). The solid line represents the median value. The undistorted incident angle ϕ versus the distorted angle ϕ' is plotted in Figure 5 (a). It can be seen that the field of view is about 186.8° compared to the specified 185°.

The data set *Subset1* is a subset of *Fisheye1* with only four images. This data set was



(a) (b) (c)

Figure 4: (a) Sample calibration image with extracted and reprojected grid points overlaid. (b) Image after rectification. (c) Radial distribution of the reprojection errors for all images contained in the set.

	Scaramuzza [10]	Mei [6]	Proposed
Fisheye1	0.434 ± 0.290	n/a	0.352 ± 0.310
Subset1	0.354 ± 0.288	0.334 ± 0.293	0.272 ± 0.310
Fisheye2	0.412 ± 0.298	n/a	0.301 ± 0.324
Parabolic	0.343 ± 0.201	0.291 ± 0.227	0.319 ± 0.300
Digiclops	0.153 ± 0.101	n/a	0.148 ± 0.102

Table 1: Comparison of the mean squared reprojection error and its standard deviation in pixels for all data sets.

produced in order to allow a comparison with the algorithm presented in [6] as further explained below.

The second data set, *Fisheye2*, was taken with a second imaging system similar to the first one. In this setup, partial occlusion was an additional challenge as not all grid points are visible in the centre area of the image. Figure 4 shows one example image from the data set containing 18 images. The sample image with the reprojected grid points overlaid is shown in Figure 4(a). Figure 4(b) represents the rectified image and (c) depicts the radial distribution of the reprojection error for the whole set.

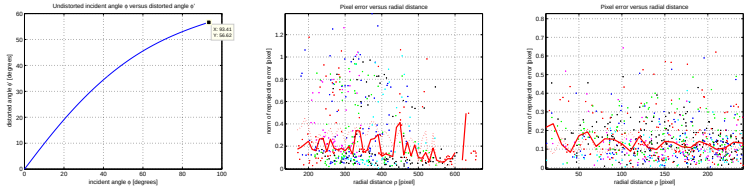
The third data set[2], *Parabolic*, was captured with a parabolic mirror with a diameter of 80mm and a focal distance of 16.7mm. The set contains 10 images.

The fourth data set[3], *Digiclops*, contains images taken with a *Digiclops* camera from *Point Grey Research*. This imaging system presents only very small distortion effects. The radial distribution of the mean squared reprojection error is illustrated in Figure 5 (b) for the third and in (c) for the fourth data set.

The proposed calibration procedure was compared to the algorithm in [10] and in [6]. Both implementations are online available as a Matlab toolbox. The toolbox from [6] does not provide an estimation for the centre of distortion and the initialisation was done manually with the results from the proposed calibration procedure. Further, the toolbox does not allow to extract grid points manually. Only the grid points of four of the images

[2]Dataset kindly provided by C. Mei, `http://www.robots.ox.ac.uk/~cmei/Toolbox.html`

[3]Dataset kindly provided from ACFR, The University of Sydney and LCR, Universidad Nacional del Sur. PAATV/UTE Projects, 2006.



(a) (b) (c)

Figure 5: (a) Incident angle ϕ versus distorted angle ϕ' for the first data set. Radial distribution of the reprojection error for the Parabolic mirror data set in (b) and for the Digiclops data set in (c).

in *Fisheye1* could be extracted successfully. To allow a comparison nevertheless, *Subset1* was created. Because of missing extrinsic parameters for the remaining images, their grid points could not be extracted. The data set *Fisheye2* could not be applied because of partial occluded grid points. The iterative procedure to find the centre of distortion in the toolbox from [10] failed for the data sets *Parabolic* as well as *Fisheye2*. They were initialised with the results from the other algorithms. For *Fisheye2*, the results from the proposed calibration procedure were used and for *Parabolic* the results from the toolbox from [6] were used.

Table 1 summarises the mean-squared reprojection errors and its standard deviation for all data sets. It can be seen that the proposed calibration procedure performs well with the miniature, low-cost, wide-angle imaging systems as in data set *Fisheye1* and *Fisheye2*. Please note that the results from *Subset1* might not be representative as the number of calibration images is not sufficient. The data sets *Parabolic* and *Digiclops* were used to show that the new model also provides good results for less complex systems (less misalignment, less distortion, and increased focal length) where the additional parameters might be redundant.

5 Summary and Conclusion

An extension to an existing calibration algorithm has been proposed in order to deal with miniature, low-cost, and wide-angle imaging systems. Experimental results with different data sets and a comparison with other algorithms showed that the proposed procedure improves the accuracy for the targeted imaging systems. The proposed camera model also performs well for less complex systems as omnidirectional mirror systems or less distorted cameras.

References

[1] J.P. Barreto and H. Araujo. Paracatadioptric camera calibration using lines. *IEEE International Conference on Computer Vision*, 2, 2003.

[2] D.C. Brown. Lens distortion for close-range photogrammetry. *Photometric Engineering*, 37(8):855–866, 1971.

[3] O.D. Faugeras, Q. Luong, and S.J. Maybank. Camera self-calibration: Theory and experiments. In *European Conference on Computer Vision*, pages 321–334, London, UK, 1992. Springer-Verlag.

[4] R. Hartley and S. B. Kang. Parameter-free radial distortion correction with centre of distortion estimation. Technical Report MSR-TR-2005-42, Microsoft Research, April 2005.

[5] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, March 2004.

[6] C. Mei and P. Rives. Single view point omnidirectional camera calibration from planar grids. In *IEEE International Conference on Robotics and Automation*, April 2007. To appear.

[7] B. Micusik and T. Pajdla. Autocalibration & 3d reconstruction with non-central catadioptric cameras. In *IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 58–65, Los Alamitos, CA, USA, 2004. IEEE Computer Society.

[8] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer-Verlag, 1999.

[9] S. Ramalingam, P. Sturm, and S. Lodha. Theory and calibration for axial cameras. In *Proceedings of the Asian Conference on Computer Vision, Hyderabad, India*, volume 1, pages 704–713, January 2006.

[10] D. Scaramuzza, A. Martinelli, and R. Siegwart. A toolbox for easily calibrating omnidirectional cameras. In *IEEE International Conference on Intelligent Robots and Systems*, pages 5695–5701, 2006.

[11] G. P. Stein. Accurate internal camera calibration using rotation, with analysis of sources of error. In *IEEE International Conference on Computer Vision*, page 230, Washington, DC, USA, 1995. IEEE Computer Society.

[12] P. Sturm and S. Ramalingam. A generic concept for camera calibration. In *Proceedings of the European Conference on Computer Vision, Prague, Czech Republic*, volume 2, pages 1–13. Springer, May 2004.

[13] X. Ying and Z. Hu. Catadioptric camera calibration using geometric invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1260–1271, 2004.

[14] Z. Zhang. A flexible new technique for camera calibration. Technical Report MSR-TR-98-71, Microsoft Research, December 1998.

Beyond Facial Expressions: Learning Human Emotion from Body Gestures

Caifeng Shan, Shaogang Gong, and Peter W. McOwan
Department of Computer Science
Queen Mary, University of London
Mile End Road, London E1 4NS, UK
{cfshan, sgg, pmco}@dcs.qmul.ac.uk

Abstract

Vision-based human affect analysis is an interesting and challenging problem, impacting important applications in many areas. In this paper, beyond facial expressions, we investigate affective body gesture analysis in video sequences, a relatively understudied problem. Spatial-temporal features are exploited for modeling of body gestures. Moreover, we present to fuse facial expression and body gesture at the feature level using Canonical Correlation Analysis (CCA). By establishing the relationship between the two modalities, CCA derives a semantic "affect" space. Experimental results demonstrate the effectiveness of our approaches.

1 Introduction

The ability to recognize affective states of a person is indispensable and important for successful interpersonal social interaction. Affective arousal modulates all nonverbal communication cues such as facial expression, body moment and posture, gesture, and tone of voice. Design and development of an automated system that can detect and interpret human affective behavior is an interesting and challenging problem [22], impacting important applications in many areas.

In computer vision, affect analysis from facial expression has been widely studied in recent years [23, 8]. However, little attention has been placed on affective body posture and gesture analysis (see Figure 1 for examples of affective body gesture), although bodily expression plays a vital role in conveying human emotional states, and the perception of facial expression is strongly influenced by the concurrently presented body language [1, 19]. This is probably due to the high variability of the emotional body posture and gesture that can be displayed. The existing studies on vision-based gesture recognition have been primarily carried out on non-affective gestures such as sign languages [27]. Emotional bodily expression has been studied in psychology and non-verbal communication. For example, Coulson [5] presented experiments on attributing six universal emotions to static body postures using computer-generated mannequin figures, and his experiments suggest that recognition from body posture is comparable to recognition from the voice, and some postures are recognized as well as facial expressions. Statistical techniques were used in [25] to determine a set of posture features in discriminating between emotions. Burgoon *et al.*[4] discussed the issue of identifying emotional states from bodily

cues for human behavior understanding. Mota and Picard [15] studied affective postures in an e-learning scenario, where the posture information was collected through a sensor chair. An affective gesture recognition system has been introduced in [26] to recognize children's emotion with intensity in context of game. Recently some tentative attempts have been made on vision-based affective body gesture analysis [2, 11]. However, there are some limitations in these studies. For example, they used very limited data (for instance, only 27 video sequences from 4 subjects were processed in [11]), and the feature extraction and representation are rather simple (for instance, the neutral and expressive frames are manually selected in [11]).

Figure 1: Examples of affective body gestures (from the FABO database [10]). From *top* to *bottom*: Fear, Joy, Uncertainty, and Surprise.

The face and the body, as part of an integrated whole, both contribute in conveying the emotional state of the individual. The studies in psychology [1, 19] suggest that the combined visual channels of facial expression and body gesture are the most informative, and their integration is a mandatory process occurring early in the human processing stream. Therefore, fusing facial expression and body gesture in video sequences provides a potential way to accomplish effective affect analysis. However, there is few efforts reported on visual affect analysis by combining face and body cues [24]. Kapoor and Picard [15] presented a multi-sensor affect recognition system for classifying the affective state of interest in children who are solving puzzles, which combines the extracted sensory information from the face videos, the senor chair (body posture), and the state of the puzzle. Balomemos *et al.*[2] attempted to analyze emotions from facial expressions and hand gestures. Recently Gunes and Piccardi [11] combined expressive face and body gestures for emotion recognition in video sequences. However, how to effectively fuse these two different modalities is still an understudied problem.

In this paper, we investigate affective body gesture analysis in video sequences. Particularly, we exploit spatial-temporal features based on space-time interest point detection [6] for representing body gestures in videos. Different from the previous studies [2, 11],

which rely on much human supervision and robust hand tracking and segmentation, our approach makes few assumptions about the observed data, such as background, occlusion and appearance. The underlining motivation is that, although two instances of the body gesture representing the same emotion may vary in terms of overall appearance and motion, due to variations across subjects or within each individual, many of the spatial-temporal features detected are similar. With regard to combining different modalities, we exploit Canonical Correlation Analysis (CCA), a powerful statistical tool that is well suited for relating two sets of signals, to fuse facial expression and body gesture at the feature level. Our motivation is that, as face and body cues are two sets of measurements for affective states, conceptually the two modalities are correlated, and their relationship can be established using CCA. CCA derives a semantic "affect" space, in which the face and body features are compatible and can be effectively fused.

Compared with the previous attempts [2, 11] in vision-based affective body gesture analysis, we make the following favorable contributions: (1) we adopt spatial-temporal features based representation for body gestures, avoiding difficulties of tracking and localization in dealing with real-world video data; (2) we present to effectively fuse two modalities at the feature level using CCA; (3)we carry out study on a much large dataset.

2 Affective Body Gesture Recognition

Recently spatial-temporal features have been investigated for event detection and behavior recognition in videos [7, 18, 16, 6, 21]. Efros *et al.*[7] introduced a motion descriptor based on optical flow measurements in a spatio-temporal volume, which was applied to recognize human action at a distance. By extending 2D rectangle features into the spatio-temporal domain, Ke *et al.*[16] presented volumetric features for event detection in videos. Laptev and Lindeberg [18] extended the spatial interest points into the spatio-temporal domain, and presented a method to detect local structures in space-time where the image values have significant local variation in both space and time. Recently Dollár *et al.*[6] proposed an alternative approach to detect sparse space-time interest points based on separable linear filters, and utilized the cuboids of spatio-temporal windowed data surrounding interest points for behavior recognition. Based on their work, Niebles *et al.*[21] more recently presented an unsupervised method for action categorization. Spatial-temporal features have been proven useful to provide a compact abstract representation of video patterns. Here we adopt spatial-temporal features to represent body gesture in videos.

2.1 Spatial-Temporal Features

We extract spatial-temporal features by detecting space-time interest points in videos. Following [6, 21], we calculate the response function by application of separable linear filters. Assuming a stationary camera or a process that can account for camera motion, the response function has the form:

$$R = (I * g * h_{ev})^2 + (I * g * h_{od})^2 \tag{1}$$

where $I(x,y,t)$ denotes images in the video, $g(x,y;\sigma)$ is the 2D Gaussian smoothing kernel, applied only along the spatial dimensions (x,y), and h_{ev} and h_{od} are a quadrature pair of 1D Gabor filters applied temporally, which are defined as $h_{ev}(t;\tau,\omega) =$

$-\cos(2\pi t\omega)e^{-t^2/\tau^2}$ and $h_{od}(t;\tau,\omega) = -\sin(2\pi t\omega)e^{-t^2/\tau^2}$. In all cases we use $\omega = 4/\tau$ [6]. The two parameters σ and τ correspond roughly to the spatial and temporal scales of the detector. Each interest point is extracted as a local maxima of the response function. As pointed out in [6], any region with spatially distinguishing characteristics undergoing a complex motion can induce a strong response, while region undergoing pure translational motion, or areas without spatially distinguishing features, will not induce a strong response.

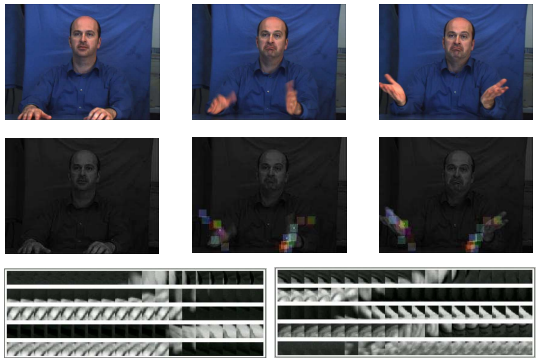


Figure 2: (Best viewed in color) Examples of spatial-temporal features extracted from videos: the first row is the original input video; the second row visualizes the cuboids extracted, where each cuboid is labeled with a different color; the third row shows some cuboids, which are flattened with respect to time.

At each detected interest point, a cuboid is extracted which contains the spatio-temporally windowed pixel values. See Figure 2 for examples of cuboids extracted. The side length of cuboids is set as approximately six times the scales along each dimension, so containing most of the volume of data that contribute to the response function at each interest point. After extracting the cuboids, the original video is discarded, which is represented as a collection of the cuboids. To compare two cuboids, different descriptors for cuboids have been evaluated in [6], including normalized pixel values, brightness gradient and windowed optical flow, followed by a conversion into a vector by flattening, global histogramming, and local histogramming. As suggested, we adopt the flattened brightness gradient as the cuboid descriptor. To reduce the dimensionality, the descriptor is projected to a lower dimensional PCA space [6]. By clustering a large number of cuboids extracted from the training data using the K-Means algorithm, we derive a library of cuboid prototypes. So each cuboid is assigned a type by mapping it to the closest prototype vector. Following [6], we use the histogram of the cuboid types to describe the video.

2.2 Recognition: SVM

we adopt the Support Vector Machine (SVM) classifier to recognize affective body gestures. SVM is an optimal discriminant method based on the Bayesian learning theory. For the cases where it is difficult to estimate the density model in high-dimensional space, the discriminant approach is preferable to the generative approach. SVM performs an implicit

mapping of data into a higher dimensional feature space, and then finds a linear separating hyperplane with the maximal margin to separate data in this higher dimensional space.

Given a training set of labeled examples $\{(x_i, y_i), i = 1, \ldots, l\}$ where $x_i \in R^n$ and $y_i \in \{1, -1\}$, a new test example x is classified by the following function:

$$f(x) = sgn(\sum_{i=1}^{l} \alpha_i y_i K(x_i, x) + b) \quad (2)$$

where α_i are Lagrange multipliers of a dual optimization problem that describe the separating hyperplane, $K(\cdot, \cdot)$ is a kernel function, and b is the threshold parameter of the hyperplane. The training sample x_i with $\alpha_i > 0$ is called the *support vector*, and SVM finds the hyperplane that maximizes the distance between the support vectors and the hyperplane. Given a non-linear mapping Φ that embeds the input data into the high dimensional space, kernels have the form of $K(x_i, x_j) = \langle \Phi(x_i) \cdot \Phi(x_j) \rangle$. SVM allows domain-specific selection of the kernel function, and the most commonly used kernel functions are the linear, polynomial, and Radial Basis Function (RBF) kernels.

3 Fusing Facial Expressions and Body Gestures

A single body gesture can be ambiguous. For example, the examples shown in the second and fourth row in Figure 1 have much similar body gesture, but the affective state they express are quite different, as shown by their facial expressions. As suggested in psychological studies [1], combining visual channels of facial expression and body gesture is a potential way to accomplish effective affect analysis.

The psychological study [19] suggests that the integration of facial expression and body gesture is a mandatory process occurring early in human processing stream. So the two modalities should be processed in a joint feature space [23], rather than fused at the decision-level. The main difficulties for the feature-level fusion are the features from different modalities may be incompatible, and the relationship between different feature spaces is unknown. Here we propose to fuse face and body cues at the feature level using CCA. Our motivation is that, as facial expression and body gesture are two sets of measurements for the affective state, conceptually they are correlated. CCA can establish their relationship, deriving a semantic "affect" space, in which the face and body features are compatible and can be effectively fused.

3.1 Canonical Correlation Analysis

CCA [13] is a statistical technique developed for measuring linear relationships between two multidimensional variables. It finds pairs of base vectors (i.e., canonical factors) for two variables such that the correlations between the projections of the variables onto these canonical factors are mutually maximized. Recently CCA has been applied to computer vision problems [3, 20, 12, 17]. Borga [3] adopted CCA to find corresponding points in stereo images. Melzer *et al.*[20] applied CCA to model the relation between an object's poses with raw brightness images for appearance-based 3D pose estimation. Harsoon *et al.*[12] presented a method using CCA to learn a semantic representation to web images and their associated text.

Given two zero-mean random variables $\mathbf{x} \in R^m$ and $\mathbf{y} \in R^n$, CCA finds pairs of directions $\mathbf{w}_x$ and $\mathbf{w}_y$ that maximize the correlation between the projections $x = \mathbf{w}_x^T \mathbf{x}$ and $y = \mathbf{w}_y^T \mathbf{y}$. The projections x and y are called *canonical variates*. More formally, CCA maximizes the function:

$$\rho = \frac{E[xy]}{\sqrt{E[x^2]E[y^2]}} = \frac{E[\mathbf{w}_x^T \mathbf{x}\mathbf{y}^T \mathbf{w}_y]}{\sqrt{E[\mathbf{w}_x^T \mathbf{x}\mathbf{x}^T \mathbf{w}_x]E[\mathbf{w}_y^T \mathbf{y}\mathbf{y}^T \mathbf{w}_y]}} = \frac{\mathbf{w}_x^T \mathbf{C}_{xy} \mathbf{w}_y}{\sqrt{\mathbf{w}_x^T \mathbf{C}_{xx} \mathbf{w}_x \mathbf{w}_y^T \mathbf{C}_{yy} \mathbf{w}_y}} \tag{3}$$

where $\mathbf{C}_{xx} \in R^{m \times m}$ and $\mathbf{C}_{yy} \in R^{n \times n}$ are the *within-set covariance matrices* of $\mathbf{x}$ and $\mathbf{y}$, respectively, while $\mathbf{C}_{xy} \in R^{m \times n}$ denotes their *between-sets covariance matrix*. A number of at most $k = \min(m,n)$ canonical factor pairs $\langle \mathbf{w}_x^i, \mathbf{w}_y^i \rangle, i = 1, \ldots, k$ can be obtained by successively solving $\arg\max_{\mathbf{w}_x^i, \mathbf{w}_y^i} \{\rho\}$ subject to $\rho(\mathbf{w}_x^j, \mathbf{w}_x^i) = \rho(\mathbf{w}_y^j, \mathbf{w}_y^i) = 0$ for $j = 1, \ldots, i-1$, i.e., the next pair of $\langle \mathbf{w}_x, \mathbf{w}_y \rangle$ are orthogonal to the previous ones.

The maximization problem can be solved by setting the derivatives of Eqn. (3), with respect to $\mathbf{w}_x$ and $\mathbf{w}_y$, equal to zero, resulting in the eigenvalue equations as:

$$\begin{cases} \mathbf{C}_{xx}^{-1} \mathbf{C}_{xy} \mathbf{C}_{yy}^{-1} \mathbf{C}_{yx} \mathbf{w}_x = \rho^2 \mathbf{w}_x \\ \mathbf{C}_{yy}^{-1} \mathbf{C}_{yx} \mathbf{C}_{xx}^{-1} \mathbf{C}_{xy} \mathbf{w}_y = \rho^2 \mathbf{w}_y \end{cases} \tag{4}$$

Matrix inversions need to be performed in Eqn. (4), leading to numerical instability if $\mathbf{C}_{xx}$ and $\mathbf{C}_{yy}$ are rank deficient. Alternatively, $\mathbf{w}_x$ and $\mathbf{w}_y$ can be obtained by computing principal angles, as CCA is the statistical interpretation of principal angles between two linear subspace [9] (see [17] for details).

3.2 Feature Fusion of Face and Body

Given $B = \{\mathbf{x} | \mathbf{x} \in R^m\}$ and $F = \{\mathbf{y} | \mathbf{y} \in R^n\}$, where $\mathbf{x}$ and $\mathbf{y}$ are the feature vectors extracted from bodies and faces respectively, we apply CCA to establish the relationship between $\mathbf{x}$ and $\mathbf{y}$. Suppose $\langle \mathbf{w}_x^i, \mathbf{w}_y^i \rangle, i = 1, \ldots, k$ are the canonical factors pairs obtained, we can use d ($1 \leq d \leq k$) factor pairs to represent the correlation information. With $\mathbf{W}_x = [\mathbf{w}_x^1, \ldots, \mathbf{w}_x^d]$ and $\mathbf{W}_y = [\mathbf{w}_y^1, \ldots, \mathbf{w}_y^d]$, we project the original feature vectors as $\mathbf{x}' = \mathbf{W}_x^T \mathbf{x} = [x_1, \ldots, x_d]^T$ and $\mathbf{y}' = \mathbf{W}_y^T \mathbf{y} = [y_1, \ldots, y_d]^T$ in the lower dimensional correlation space, where x_i and y_i are uncorrelated with the previous pairs x_j and $y_j, j = 1, \ldots, i-1$. We then combine the projected feature vector $\mathbf{x}'$ and $\mathbf{y}'$ to form the new feature vector as

$$\mathbf{z} = \begin{pmatrix} \mathbf{x}' \\ \mathbf{y}' \end{pmatrix} = \begin{pmatrix} \mathbf{W}_x^T \mathbf{x} \\ \mathbf{W}_y^T \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{W}_x & 0 \\ 0 & \mathbf{W}_y \end{pmatrix}^T \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \tag{5}$$

This fused feature vector effectively represents the multimodal information in a joint feature space for affect analysis.

4 Experiments

There are several facial expression databases in affective-computing community, but few databases containing affective body gestures. Gunes and Piccardi [10] recently collected a bimodal face and body gesture database (FABO), which consists of facial expression and body gesture recorded simultaneously. The database includes 23 subjects in age from

18 to 50 years, of which 12 were female, 23 were from Europe, 2 from Middle East, 3 from Latin America, 7 from Asia, and 1 from Australia. In total there are around 1900 videos. Examples of the video sequences are shown in Figure 1. In our experiments, we selected 262 videos of seven emotions (Anger, Anxiety, Boredom, Disgust, Joy, Puzzle, and Surprise) from 23 subjects. Gunes and Piccardi [11] reported some preliminary results on this database, but they only used 54 videos from 4 subjects.

4.1 Affective Body Gesture Recognition

To evaluate the algorithms' generalization ability, we adopted a 5-fold cross-validation test scheme in all recognition experiments. That is, we divided the data set randomly into five groups with roughly equal number of videos, and then used the data from four groups for training and the left group for testing; the process was repeated five times for each group in turn to be tested. We report the average recognition rates here. In all experiments, we set the soft margin C value of SVMs to infinity so that no training error was allowed. Meanwhile, each training and testing vector was scaled to be between -1 and 1. In our experiments, the RBF kernel always provided the best performance, so we report the performance of the RBF kernel. With regard to the hyper-parameter selection of RBF kernels, as suggested in [14], we carried out grid-search on the kernel parameters in the 5-fold cross-validation. The parameter setting producing the best cross-validation accuracy was picked. We used the SVM implementation in the publicly available machine learning library SPIDER [1] in our experiments.

We compare the SVM classifier with the 1-nearest neighbor classifier used in [6] for affective body gesture recognition. The average recognition rates of SVM and 1-nearest neighbor classifier are 72.6% and 68.6% respectively. We plot the confusion matrices of the two classifier in Figure 3. It can observed that the SVM classifier slightly outperforms the 1-nearest neighbor classifier.

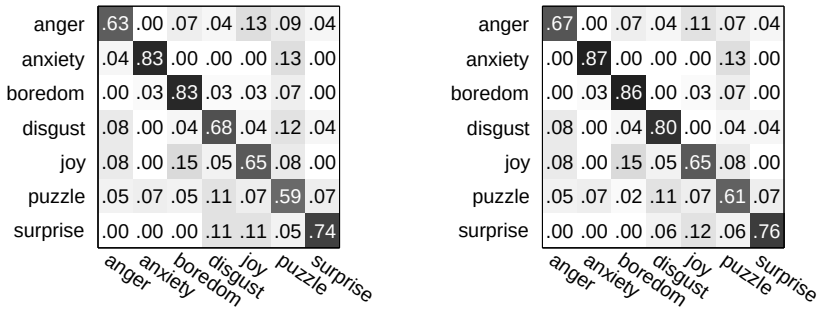


Figure 3: Confusion matrices of affective body gesture recognition with the 1-nearest neighbor classifier (*left*) and the SVM classifier (*right*).

[1] *http://www.kyb.tuebingen.mpg.de/bs/people/spider/index.html*

4.2 Emotion Recognition by Fusing Face and Body Cues

In the FABO database, video sequences were recorded simultaneously using two video cameras, one is for capturing the facial expression only and the other for capturing upper-body movements. We extracted the spatial-temporal features from the face video and the body video, and then fuse the two modalities at the feature level using CCA. We first report the classification performance based on facial cues only. The confusion matrices of the two classifiers are shown in Figure 4, and the recognition rates of SVM and 1-nearest neighbor classifier are 79.2% and 74.8% respectively. We can see that the emotion classification based on facial expressions is better than that of body gesture. This is possibly because there are much variation in affective body gestures.

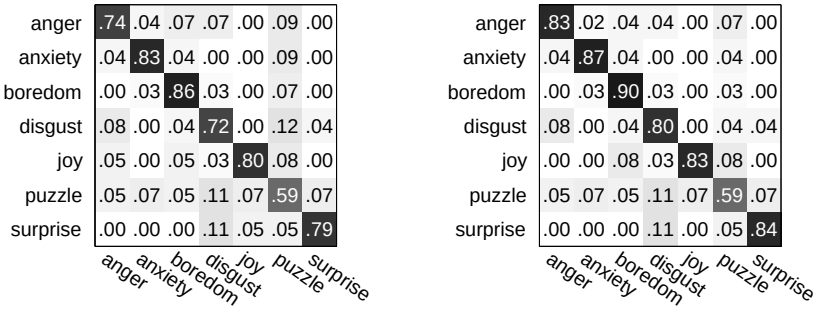


Figure 4: Confusion matrices of facial expression recognition with the 1-nearest neighbor classifier (*left*) and the SVM classifier (*right*).

We then fused facial expression and body gesture at the feature level using CCA. Different numbers of CCA factor pairs can be used to project the original face and body feature vectors to a lower dimensional CCA feature space, and the recognition performance varies with the dimensionality of the projected CCA features. We report the best result obtained here. We compared the CCA feature fusion with another three feature fusion methods: (1) Direct feature fusion, that is, concatenating the original body and face features to derive a single feature vector; (2) PCA feature fusion: the original body and face features are first projected to the PCA space respectively, and then the PCA features are concatenated to form the single feature vector. In our experiments, all principle components were kept. (3) PCA+LDA feature fusion: for each modality, the derived PCA features are further projected to the discriminant LDA space; the LDA features are then combined to derive the single feature vector. We report the experimental results of different feature fusion schemes in Table 1. The confusion matrices of the CCA feature fusion and the direct feature fusion are shown in Figure 5. We can see that the presented CCA feature fusion provides best recognition performance. This is because CCA captures the relationship between the feature sets in different modalities, and the fused CCA features effectively represent information from each modality.

Feature Fusion	CCA	Direct	PCA	PCA+LDA
Recognition Rate	88.5%	81.9%	82.3%	87.8%

Table 1: Experimental results of affect recognition by fusing body and face cues.

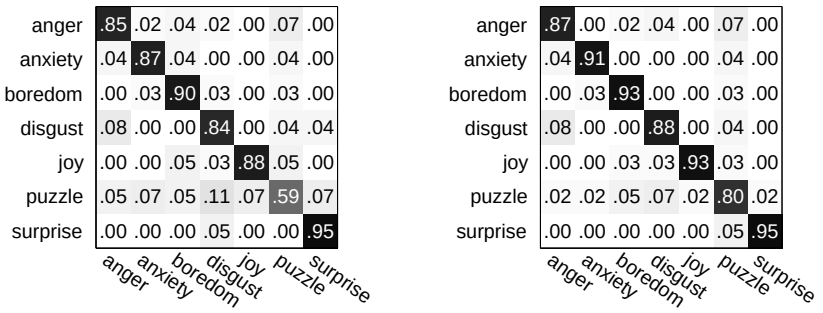


Figure 5: Confusion matrices of affect recognition by fusing facial expression and body gesture. (*left*) Direct feature fusion; (*right*) CCA feature fusion.

5 Conclusions and Discussions

In this paper, we investigate affective body gesture analysis in videos, a relatively understudied problem. Spatial-temporal features are exploited for modeling of body gestures. We also present to fuse facial expression and body gesture at the feature level using Canonical Correlation Analysis. The current spatial-temporal features based video description does not consider the position relations of cuboids detected. By including the relative position information between the cuboid types, the representation will be much more discriminative. This will be studied in our future work.

References

[1] N. Ambady and R. Rosenthal. Thin slices of expressive behaviour as predictors of interpersonal consequences: A meta-analysis. *Psychological Bulletin*, 111(2):256–274, February 1992.

[2] T. Balomenos, A. Raouzaiou, S. Ioannou, A. Drosopoulos, K. Karpouzis, and S. Kollias. Emotion analysis in man-machine interaction systems. In *Machine Learning for Multimodal Interaction, LNCS 3361*, pages 318–328, 2005.

[3] M. Borga. *Learning Multidimensional Signal Processing*. PhD thesis, Linkoping University, SE-581 83 Linkoping, Sweden, 1998. Dissertation No 531.

[4] J. K. Burgoon, M. L. Jensen, T. O. Meservy, J. Kruse, and J. F. Nunamaker. Augmenting human identification of emotional states in video. In *International Conference on Intelligent Data Analysis*, 2005.

[5] M. Coulson. Attributing emotion to static body postures: Recognition accuracy, confusions, and viewpoint dependence. *Journal of Nonverbal Behavior*, 28(2):117–139, Summer 2004.

[6] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, pages 65–72, 2005.

[7] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *IEEE International Conference on Computer Vision (ICCV)*, pages 726–733, 2003.

[8] B. Fasel and J. Luettin. Automatic facial expression analysis: a survey. *Pattern Recognition*, 36:259–275, 2003.

[9] G. H. Golub and H. Zha. The canonical correlations of matrix pairs and their numerical computation. Technical report, 1992.

[10] H. Gunes and M. Piccardi. A bimodal face and body gesture database for automatic analysis of human nonverbal affective behavior. In *International Conference on Pattern Recognition (ICPR)*, volume 1, pages 1148–1153, 2006.

[11] H. Gunes and M. Piccardi. Bi-modal emotion recognition from expressive face and body gestures. *Journal of Network and Computer Applications*, 2007.

[12] D. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical correlation analysis; an overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, 2004.

[13] H. Hotelling. Relations between two sets of variates. *Biometrika*, 8:321–377, 1936.

[14] C.-W. Hsu, C.-C. Chang, and C.-J. Lin. A practical guide to support vector classification. Technical report, Taipei, 2003.

[15] A. Kapoor and R. W. Picard. Multimodal affect recognition in learning environments. In *ACM International Conference on Multimedia*, 2005.

[16] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *IEEE International Conference on Computer Vision (ICCV)*, pages 166–173, 2005.

[17] T-K. Kim, J. Kittler, and R. Cipolla. Learning discriminative canonical correlations for object recognition with image sets. In *European Conference on Computer Vision (ECCV)*, pages 251–262, 2006.

[18] I. Laptev and T. Lindeberg. Space-time interest points. In *IEEE International Conference on Computer Vision (ICCV)*, pages 432–439, 2003.

[19] H. Meeren, C. Heijnsbergen, and B. Gelder. Rapid perceptual integration of facial expression and emotional body language. *Procedings of the National Academy of Sciences of USA*, 102(45):16518–16523, November 2005.

[20] T. Melzer, M. Reiter, and H. Bischof. Appearance models based on kernel canonical correlation analysis. *Pattern Recognition*, 39(9):1961–1973, 2003.

[21] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. In *British Machine Vision Conference (BMVC)*.

[22] M. Pantic, A. Pentland, Nijholt A., and T.S. Huang. Human computing and machine understanding of human behavior: A survey. In T.S. Huang, A. Nijholt, M. Pantic, and A. Pentland, editors, *Artificial Intelligence for Human Computing*, volume 4451, pages 47–71. Springer, 2007.

[23] M. Pantic and L. Rothkrantz. Toward an affect-sensitive multimodal human-computer interaction. In *Proceeding of the IEEE*, volume 91, pages 1370–1390, 2003.

[24] M. Pantic, N. Sebe, J. Cohn, and T. Huang. Affective multimodal human-computer interaction. In *ACM International Conference on Multimedia*, pages 669–676, 2005.

[25] P. Ravindra De Silva and N. Bianchi-Berthouze. Modeling human affective postures: an information theoretic characterizatino of posture features. *Computer Animation and Virtual Worlds*, 15:169–276, 2004.

[26] P. Ravindra De Silva, M. Osano, and A. Marasinghe. Towards recognizing emotion with affective dimensions through body gestures. In *IEEE International Conference on Automatic Face & Gesture Recognition (FG)*, 2006.

[27] Y. Wu and T. Huang. Human hand modeling, analysis and animation in the context of human computer interaction. *IEEE Signal Processing Magazine*, 18(3):51–60, May 2001.

the DIARETDB1 diabetic retinopathy database and evaluation protocol

T. Kauppi[1] V. Kalesnykiene[2] J.-K. Kamarainen[1,3] L. Lensu[1] I. Sorri[2]
A. Raninen[2] R. Voutilainen[2] J. Pietilä[5]
H. Kälviäinen[1] H. Uusitalo[2]
[1]Machine Vision and Pattern Recognition Research Group,
Lappeenranta University of Technology
[2]Department of Ophthalmology, Faculty of Medicine,
University of Kuopio
[3]Centre for Vision, Speech and Signal Processing,
University of Surrey
[5]Perimetria Ltd., Helsinki, Finland

Abstract

Automatic diagnosis of diabetic retinopathy from digital fundus images has been an active research topic in the medical image processing community. The research interest is justified by the excellent potential for new products in the medical industry and significant reductions in health care costs. However, the maturity of proposed algorithms cannot be judged due to the lack of commonly accepted and representative image database with a verified ground truth and strict evaluation protocol. In this study, an evaluation methodology is proposed and an image database with ground truth is described. The database is publicly available for benchmarking diagnosis algorithms. With the proposed database and protocol, it is possible to compare different algorithms, and correspondingly, analyse their maturity for technology transfer from the research laboratories to the medical practice.

1 Introduction

Diabetes has become one of the rapidly increasing health threats worldwide [15]. Only in Finland, there are 30 000 people diagnosed to the type 1 maturity onset diabetes in the young, and 200 000 people diagnosed to the type 2 latent autoimmune diabetes in adults [4]. In addition, the current estimate predicts that there are 50 000 undiagnosed patients [4]. Proper and early treatment of diabetes is cost effective since the implications of poor or late treatment are very expensive. In Finland, diabetes costs annually 505 million euros for the Finnish health care, and 90% of the care cost arises from treating the complications of diabetes [5]. These alarming facts promote the study of automatic diagnosis methods for screening over large populations.

Fundus imaging has an important role in diabetes monitoring since occurrences of retinal abnormalities are common and their consequences serious. However, since the eye fundus is sensitive to vascular diseases, fundus imaging is also considered as a candidate

for non-invasive screening. The success of this type of screening approach depends on accurate fundus image capture, and especially on accurate and reliable image processing algorithms for detecting the abnormalities.

Numerous algorithms have been proposed for fundus image analysis by many research groups [9, 6, 18, 11, 13]. However, it is impossible to judge the accuracy and reliability of the approaches because there exists no commonly accepted and representative fundus image database and evaluation protocol. With a widely accepted protocol, it would be possible to evaluate the maturity and state-of-the-art of the current methods, i.e., produce the achieved sensitivity and selectivity rates. For example, commonly accepted strict guidelines for the evaluation of biometric authentication methods, such as the FERET and BANCA protocols for face recognition methods [12, 2], have enabled the rapid progress in that field, and the same can be expected in medical image processing related to diabetic retinopathy detection.

The main contribution of this work is to report a publicly available diabetic retinopathy database, DIARETDB1, containing the ground truth collected from several experts and a strict evaluation protocol. The protocol is demonstrated with a baseline method included to the available toolkit. This study provides the means for the reliable evaluation of automatic methods for detecting diabetic retinopathy.

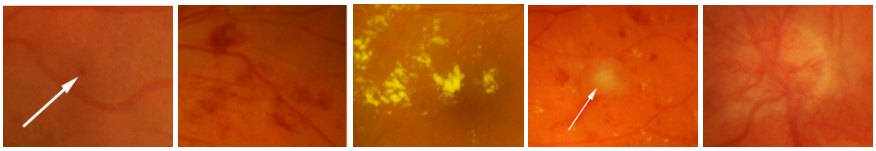
2 Diabetic retinopathy

In the type 1 diabetes, the insulin production in the pancreas is permanently damaged, whereas in the type 2 diabetes, the person is suffering from increased resistance to insulin. The type 2 diabetes is a familial disease, but also related to limited physical activity and lifestyle [15]. The diabetes can cause abnormalities in the retina (diabetic retinopathy), kidneys (diabetic nefropathy), and nervous system (diabetic neuropathy) [10]. The diabetes is also a major risk factor in cardiovascular diseases [10].

The diabetic retinopathy is a microvascular complication of diabetes, causing abnormalities in the retina, and in the worst case, blindness. Typically there are no salient symptoms in the early stages of diabetic retinopathy, but their number and severity predominantly increase with time. The diabetic retinopathy typically begins as small changes in the retinal capillaries. The first detectable abnormalities are mircroaneurysms (Ma) shown in Fig. 1(a), which are local distensions of the retinal capillary and when ruptured, cause intraretinal hemorrhage (H) shown in Fig. 1(b). The disease severity is classified as mild non-proliferative diabetic retinopathy when the first apparent microaneurysms appear in the retina [17]. With time, the retinal edema and hard exudates (He) shown in Fig. 1(c) appear because of the increased permeability of the capillary walls. The hard exudates are lipid formations leaking from these weakened blood vessels. This state of the retinopathy is called moderate non-proliferative diabetic retinopathy [17]. However, if the above-mentioned abnormalities appear in the central vision area (macula), the condition is called diabetic maculopathy [15]. As the retinopathy advances, the blood vessels become obstructed which causes microinfarcts in the retina. These microinfarcts are called soft exudates (Se) shown in Fig. 1(d). When a significant number of intraretinal hemorrhages, soft exudates, or intraretinal microvascular abnormalities are encountered, the state of the retinopathy is defined as severe non-proliferative diabetic retinopathy [17].

The severe non-proliferative diabetic retinopathy can quickly turn into proliferative di-

abetic retinopathy when extensive lack of oxygen causes the development of new fragile blood vessels [17]. This is called neovascularisation shown in Fig. 1(e), which is a serious eye sight threatening state. The proliferative diabetic retinopathy may cause sudden loss in visual acuity, or even permanent blindness due to vitreous hemorrhage or tractional detachment of the central retina. After the diabetic retinopathy has been diagnosed, regular monitoring is needed due to the progressive nature of the disease. However, broad pre-emptive screenings cannot be performed due to the fact that the fundus image examination requires attention of medical experts. For the screening, reliable automatic image processing methods must be developed.



(a) (b) (c) (d) (e)

Figure 1: Abnormal findings in the eye fundus caused by the diabetic retinopathy (best viewed in colour): (a) microaneuryms (marked with an arrow); (b) hemorrhages; (c) hard exudates; (d) soft exudate (marked with an arrow); (e) neovascularization.

2.1 Current evaluation practises

In medical diagnosis, the medical input data is usually classified into two classes, where the disease is either present or absent. The classification accuracy of the diagnosis is assessed using the sensitivity and specificity measures. Following the practises in the medical research, the fundus images related to the diabetic retinopathy are evaluated by using sensitivity and specificity per image basis. Sensitivity is the percentage of abnormal funduses classified as abnormal, and specificity is the percentage of normal fundus classified as normal by the screening. The higher the sensitivity and specificity values, the better the diagnosis. Sensitivity and specificity are computed as [16]:

$$\text{sensitivity (SN)} = \frac{T_P}{T_P + F_N}, \ \text{specificity (SP)} = \frac{T_N}{T_N + F_P} \tag{1}$$

where T_P is the number of abnormal fundus images found as abnormal, T_N is the number of normal fundus images found as normal, F_P is the number of normal fundus images found as abnormal (false positives) and F_N is the number of abnormal fundus images found as normal (false negatives). Sensitivity and specificity are also referred to as the true positive rate (TPR) and true negative rate (TNR), respectively.

3 Evaluation database

A necessary tool for reliable evaluations and comparisons of medical image processing algorithms is a database of dedicatedly selected high-quality medical images which are representatives of the problem and have been verified by experts. In addition, information about the medical findings, the ground truth, must accompany the image data. An accurate

algorithm should take the images as input, and produce output which is consistent with the ground truth. In the evaluation, the consistency is measured, and algorithms can be compared based on these performance metrics. In the following, we describe the images and ground truth for the diabetic retinopathy database DIARETDB1.

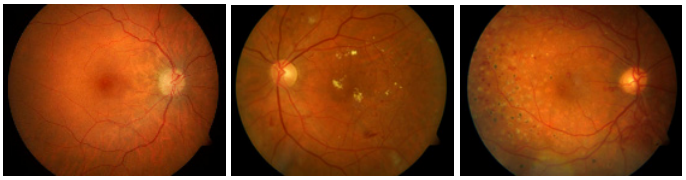
3.1 Fundus images

The database consists of 89 colour fundus images of which 84 contain at least mild non-proliferative signs (Ma) of the diabetic retinopathy (two examples shown in Figs. 2(b) and 2(c)), and 5 are considered as normal which do not contain any signs of the diabetic retinopathy according to all experts participated in the evaluation (an example shown in Fig. 2(a)). The images were taken in the Kuopio university hospital. The images were selected by the medical experts, but their distribution does not correspond to any typical population, i.e., the data is biased and no a priori information can be devised from it. The diabetic retinopathy abnormalities in the database are relatively small, but they appear near the macula which is considered to threaten the eyesight. Images were captured with the same 50 degree field-of-view digital fundus camera [1] with varying imaging settings and without preprocessing. The intensity of the camera flash was controlled by the photographer, but other settings such as shutter speed, aperture, and gain were controlled by the fundus camera. The images contain a varying amount of imaging noise, but the optical aberrations (dispersion, transverse and lateral chromatic, spherical, field curvature, coma, astigmatism, distortion) and photometric accuracy (colour or intensity) are the same. Therefore, the system induced photometric variance over the visual appearance of the different retinopathy findings can be considered as small. In the absence of a well-founded procedure for camera characterisation, the data correspond to a good (not necessarily typical nor the most general) practical situation, where the images are comparable, and can be used to evaluate the general performance of diagnostic methods. The general performance corresponds to the situation where no calibration is performed (actual physical measurement values cannot be recovered), but where the images correspond to commonly used imaging conditions, i.e., the conditions encountered in a single hospital. This data set is referred to as "calibration level 1 fundus images". A data set taken with several fundus cameras containing different amounts of imaging noise and optical aberrations is referred to as "calibration level 0 fundus images". To publish an ultimate tool for the evaluation of diabetic retinopathy, which is the research group's main objective, it is necessary to study the different calibration levels. Therefore, the current database was not aimed to be statistically representative, but as one step in the development process.

3.2 Ground truth

Independent markings from 4 medical experts were collected by using a software tool provided for image annotation. The computer displays used in the collection process were not calibrated. A person with medical education and solid experience in ophthalmology was considered as an expert. The experts were asked to mark the areas related to the microaneurysms, hemorrhages, and hard and soft exudates. The experts were instructed to avoid marking the findings so that the borders of the marked areas contain any pixels belonging to the finding. The experts were further instructed to report their confidence and

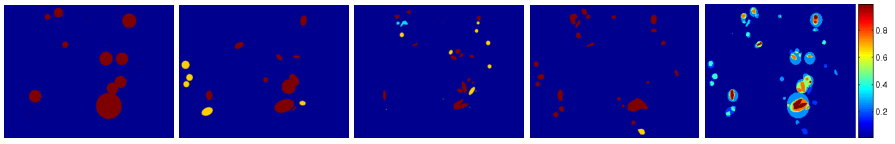
[1] ZEISS FF 450^{plus} fundus camera with Nikon F5 digital camera



(a) (b) (c)

Figure 2: Examples of DIARETDB1 fundus images(best viewed in colours): (a) normal fundus, (b) abnormal fundus, and (c) abnormal fundus after treatment by photocoagulation.

especially annotate the single most representative point for each finding. The ground truth confidence levels defined by the medical experts, $\{< 50\%,\ > 50\%,\ 100\%\}$, represented the certainty of the decision that a marked finding is correct. Medical experts had varying certainty for less distinguishable findings and therefore inexact confidence intervals were approved for such findings. The experts were taught to use the image annotation tool, but they were not instructed how to make the annotations to prevent a biased scheme; the medical experts learnt their own best practises. The uninstructed collection process caused significant differences between the medical experts as can be seen in Fig. 3 for image in Fig. 2(b). Therefore, it was not possible to use the expert information as such as the ground truth. However, using the original data the expert knowledge was fused for a better spatial accuracy and suppression of outliers. The fusion was performed on a pixel basis using the reported confidence levels. Several different approaches for fusing the markings



(a) (b) (c) (d) (e)

Figure 3: Expert marked hemorrhages for the same image (colour decodes the ground truth confidence): (a) expert 1; (b) 2; (c) 3; (d) 4; (e) mean.

are possible, e.g., voting, minimum, maximum and the sum (average) of confidences. The first three provide binary classifications, but the normalised average provides values in the range $[0,1]$ (see Fig. 3(e)). It should be noted that the markings do not provide any absolute ground truth of the findings, but reveal how the medical experts analyse and interpret the retinopathy from the digital fundus images. Not to discard any information, the approach using the average was selected since it provides a linear confidence scale, and in the evaluation, the confidence level can be fixed to one or several values. In the DIARETDB1, the confidence level is fixed to $conf_{GT} = 0.75$.

3.3 Training and test set

The 89 images were manually assigned into categories representing the progressive states of retinopathy: normal (27 images), mild (7 images), moderate and severe non-proliferative

(28 images), and profilerative (27 images). Using the categories, the images were divided into the representative training (28 images) and test sets (61 images).

In the training set with $conf_{GT} = 0.75$, 18 images contain hard exudates, 6 soft exudates, 19 microaneurysms, and 21 hemorrhages. In the test set with $conf_{GT} = 0.75$, 20 images contain hard exudates, 9 soft exudates, 20 microaneurysms, and 18 hemorrhages. Note that a single image may contain several finding types.

4 Evaluation protocol

The training and test images, expert ground truth, a baseline method, Matlab functionality for computing performance measures and more detailed technical documentation are publicly available at the DIARETDB1 web page (`http://www.it.lut.fi/project/imageret/`). Research groups investigating diabetic retinopathy detection methods are encouraged to report their results according to the DIARETDB1 evaluation protocol. The performance measures for which the functionality is provided are defined next.

4.1 Performance measures

In the literature, the sensitivity and specificity values are typically reported since they correspond to the current medical practice and have straightforward interpretations in the medical terms. Sensitivity value depends on the diseased population and specificity on the healthy population (see Eq. 1). These values provide the means for analysing how many diseased and how many healthy patients are correctly diagnosed with a provided method. From the method comparison point of view, however, these two values are not feasible since the two distributions overlap and the true accuracy is always a trade-off.

In our evaluation protocol we have selected two evaluation principles: 1) the evaluation is image-based and 2) is done separately to different diabetic retinopathy findings (Sec. 2). The first principle is justified by the fact that it corresponds to the medical practice where decisions are “patient-based”. Spatial area (pixel-wise) based evaluation can be useful in method development, but the problem itself is always image-based. The second principle is due to the practical fact that most researchers concentrate only on one or several finding types and a practically useful method does not necessarily need to detect all findings. The image acquisition process affects the evaluation protocol only through the ground truth, i.e., the marking accuracy of experts depends on the quality of image acquisition.

4.1.1 ROC

For a proper comparison the sensitivity and specificity values must be combined into a form which can describe the behavior over different combinations of the values. Receiving operating curve (ROC) is a natural selection due to its popularity and proven applicability in similar computer vision tasks, such as face recognition [7], object class recognition [3] and medical research [8]. The ROC provides a graphical representation for sensitivity (TPR) and 1-specificity (FPR) trade off. The ROC curve provides the means for the optimal analysis when the problem is to find the best method parameters for the task or compare performances irrespective to operating conditions.

In our evaluation we adapted the practises from [3], where each method is required to provide a score for each test image. A high score corresponds to a high probability that a finding is present in the image. By manipulating the provided scores the ROC curve can be automatically generated.

4.1.2 Weighted error rate (WER)

The ROC curve is a reliable method for method comparisons, but often method ranking is also needed, and then, single valued measures must be used. Single valued measures should be derived from the corresponding ROC curve, e.g. by computing an equal error rate (EER) ($TPR = TNR$) or a total area under roc curve (AUC). Here we prefer the more interpretable EER. The EER measure however assumes equal penalties for the both false positives and negatives, which is not typically the case in the medical diagnosis. Therefore, we adapt a more versatile measure utilised in [12, 2], where the two measures, sensitivity (SN) and specificity (SP), are combined to a weighted error rate defined as

$$WER(R) = \frac{FPR + R \cdot FNR}{1+R} = \frac{(1-SP) + R \cdot (1-SN)}{1+R} \quad . \tag{2}$$

In (2) $R = \frac{C_{FNR}}{C_{FPR}}$ is the cost ratio between FPR and FNR ($R = 1$ corresponds to the equal penalty for the both). In the DIARETDB1 protocol we adopted the following measures [2]: $\mathbf{WER}(\mathbf{10^{-1}})$ (FNR is an order of magnitude less harmful), $\mathbf{WER}(\mathbf{1})$ (FPR and FNR are equally harmful) and $\mathbf{WER}(\mathbf{10})$ (FNR is an order of magnitude more harmful). These measures are computed from the nearest true points on the ROC without interpolation.

5 Evaluation example

In this section we present example evaluation according to the DIARETDB1 protocol. These reproducible results were computed using a simple baseline method which is included to the DIARETDB1 toolkit available at the web site.

5.1 Baseline method

Our method is based on the principle that different findings can be distinguished and detected based only on their photometric information, i.e. colour. We adapt the successful colour locus based face detection by Hadid et al. [1] to our multi-class diabetic retinopathy detection. The approach is justified by the fact that the photometric characteristics (illumination, camera and optics) remain the same in the DIARETDB1 images (calibration level 1). The colour variation should resemble the normal variation within the finding type and between different individuals.

The method utilises two colour channels (e.g. R and G) without intensity component (e.g. normalisation by $R+G+B$). It should be noted that no particular improvement can be achieved by changing the colour space [1], and therefore, RGB was used. A colour locus for each finding type, F_i, is defined by forming their colour histograms $h_{F_i}(r,g)$. The histograms are computed from the intensity normalised pixel colours at the neighborhood (8x8) of the most representative points marked by the experts. By using the colour histograms of findings, $h_{F_i}(r,g)$, and a test image itself, $h_{total}(r,g)$, Schwerdt and Crowley [14] have derived a formula for the Bayesian decision rule to classify a pixel with

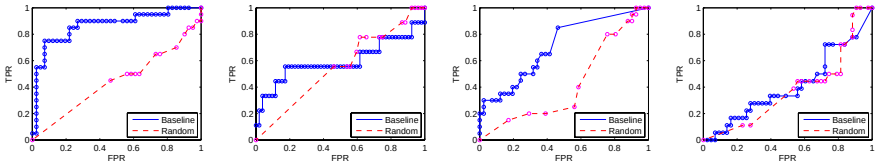
color (r,g) to one of the finding classes. The formula reduces to the histogram ratio of finding and test image:

$$p(F_i|r,g) = \frac{h_{F_i}(r,g)}{h_{total}(r,g)} \tag{3}$$

An optimal posterior threshold for every finding type was defined by manual cross-selection and finally the sum of pixels having higher or equal value to the posterior threshold was used as the image based score.

5.2 Results

Evaluation results, ROC and WER, for the baseline method are shown in Fig. 4 and Table 1, respectively. For a better visualisation also ROC curves of random classification (random score for each test image) are plotted. It is clear that the method performs moderately for the hard exudates while other findings are quite poorly detected. These results can however be used as the baseline which all reported methods should outperform. Only the WER values should be reported, but in Table 1 also the corresponding false positive and negative rates are given.



(a) (b) (c) (d)

Figure 4: ROC curves for the DIARETDB1 baseline method: a) hard exudates; b) soft exudates; c) microaneurysms; d) hemorrhages.

Table 1: The DIARETDB1 baseline performance measures.

Baseline method									
	FPR	FNR	**WER**			FPR	FNR	**WER**	
He	0.8049	0	0.0732	R = 0.1	Se	1.0000	0	0.0909	R = 0.1
	0.0732	0.2500	0.1616	R = 1		0.1731	0.4444	0.3088	R = 1
	0.0244	0.4500	0.0631	R = 10		0	0.8889	0.0808	R = 10
Ma	0.4634	0.1500	0.1785	R = 0.1	H	0.8837	0.2222	0.2824	R = 0.1
	0.4634	0.1500	0.3067	R = 1		0.1628	0.8333	0.4981	R = 1
	0	0.8000	0.0727	R = 10		0	1.0000	0.0909	R = 10

6 Discussion and future research

The development of medical image processing methods to a mature level where they are ready to be transferred from the research laboratories to medical practise requires properly designed benchmarking databases and protocols. The method testing must correspond to the strict regulations in the medical treatment and medicinal research. Medical image processing is not different from the medical practice in that sense.

We proposed a step towards standardized evaluation of methods for detecting findings of diabetic retinopathy by introducing the publicly available DIARETDB1 database and evaluation protocol. The quality and size of the database can be improved but already now the DIARETDB1 corresponds to the situation in practice very well. In the future, however, we will continue to develop the database and evaluation methodology. The following development steps will be taken: 1) a predefined set of instructions are defined for the experts to prevent the free form description, and thus, allow control over subjective interpretations and acquire spatially more accurate ground truth. 2) the effect of display calibration for the experts will be evaluated 3) location of normal findings will be added to the ground truth and their evaluation to the protocol

7 Conclusion

An image database, ground truth and evaluation methodology were proposed for evaluating and comparing methods for automatic detection of diabetic retinopathy. All data, a baseline method and evaluation functionality (toolkit) are publicly available at the DIARETDB1 web site (IMAGERET[2], `http://www.it.lut.fi/project/imageret/`). DIARETDB1 provides a unified framework for benchmarking the methods, but also points out clear deficiencies in the current practice in the method development. The work will continue and the research group's main objective is to publish an ultimate tool for the evaluation of diabetic retinopathy detection methods. The tool will provide accurate and reliable information of method performance to estimate their maturity before starting the technology transfer from the research laboratories to practice and industry.

References

[1] A. Hadid A, M. Pietikäinen, and B. Martinkauppi B. Color-based face detection using skin locus model and hierarchical filtering. In *Proc. 16th International Conference on Pattern Recognition*, pages 196–200.

[2] E. Bailliere, S. Bengio, F. Bimbot, M. Hamouz, J. Kittler, J. Mariethoz, J. Matas, K. Messer, V. Popovici, F. Poree, B. Ruiz, and J.P. Thiran. The BANCA database and evaluation protocol. In *Proc. of the Int. Conf. on Audio- and Video-Based Biometric Person Authentication*, pages 625–638, 2003.

[3] Mark Everingham and Andrew Zisserman. The pascal visual object classes challenge 2006 (voc2006) results. Workshop in ECCV06, May. Graz, Austria.

[4] Finnish Diabetes Association. Development programme for the prevention and care of diabetes, 2001. ISBN 952 5301-13-3.

[5] Finnish Diabetes Association. Programme for the prevention of type 2 diabetes in Finland, 2003. ISBN 952-5301-36-2.

[6] Alan D. Fleming, Sam Philip, Keith A. Goatman, John A. Olson, and Peter F. Sharp. Automated microaneurysm detection using local contrast normalization and

[2] Supported from the FinnWell technology program (40430/05, 40039/07) of the Finnish Funding Agency for Technology and Innovation, Tekes.

local vessel detection. *IEEE Transactions in Medical Imaging*, 25(9):1223– 1232, September 2006.

[7] P.J. Grother, R.J. Micheals, and P.J. Phillips. Face recognition vendor test 2002 performance metrics. In *Proc. of the. 4th Int. Conf. on Audio- and Video-based Biometric Person Authentication*, 2003.

[8] Thomas A. Lasko, Jui G. Bhagwat, Kelly H. Zou, and Lucilla Ohno-Machado. The use of receiver operating characteristic curves in biomedical informatics. *Journal of Biomedical Informatics*, 38:404–415, 2005.

[9] M. Niemeijer, B. van Ginneken, J. Staal, M. S. A. Suttorp-Schulten, and N. D. Abramoff. Automatic detection of red lesion in digital color fundus photographs. *IEEE Transactions on Medical Imaging*, 24(5):584–592, May 2005.

[10] M. Niemi and K. Winell. Diabetes in Finland, prevalence and variation in quality of care. Kirjapaino Hermes Oy, Tampre, Finland, 2006.

[11] Alireza Osareh, Majid Mirmehdi, Barry Thomas, and Richard Markham. Classification and localization of diabetic-related eye disease. In *Proc. of 7th European Conference on Computer Vision (ECCV)*, pages 502–516, 2002.

[12] P.J. Phillips, H. Moon, S.A. Rizvi, and P.J. Rauss. The FERET evaluation methodology for face-recognition algorithms. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(10), 2000.

[13] C. I. Sànchez, R. Hornero, M. I .Lòpez, and J. Poza. Retinal image analysis to detect and quantify lesions associated with diabetic retinopathy. In *Proceedings of the 26th Annual International Confrence of the IEEE Engineering in Medicine and Biology Society (EMBS)*, pages 1624–1627, San Francisco, CA, USA, September 2004.

[14] K. Schwerdt and J.L. Crowley. Robust face tracking using color. In *Proceedings of 4th IEEE International Conference on Automatic Face and Gesture Recognition*, 2000.

[15] Gunvor von Wendt. *Screening for diabetic retinopathy: Aspects of photographic methods*. PhD thesis, Karolinska Institutet, 2005.

[16] T. Walter, J.-C. Klein, P. Massin, and A. Erginay. A contribution of image processing to the diagnosis of diabetic retinopathy - detection of exudates in color fundus images of the human retina. *IEEE Transactions on Medical Imaging*, 21:1236–1243, October 2002.

[17] C. P. Wilkinson, Frederick L. Ferris, Ronald E. Klein, Paul P. Lee, Carl David Agardh, Matthew Davis, Diana Dills, Anselm Kampik, R. Pararajasegaram, and Juan T. Verdaguer. Proposed international clinical diabetic retinopathy and diabetic macular edema disease severity scales. *Ophthalmology*, 10(9):1677–1682, September 2003.

[18] Xiaohoui Zhang and Opas Chutape. A SVM approach for detection of hemorrhages in background diabetic retinopathy. In *Proceedings of International Joint Confrence on Neural Networks*, pages 2435–2440, Montreal and Canada, July 2005.

Hilbert-Huang Transform-based Local Regions Descriptors

Dongfeng Han, Wenhui Li, Wu Guo
Computer Science and Technology,
Key Laboratory of Symbol Computation and
Knowledge Engineering of the Ministry of Education,
Jilin University, Changchun, P. R. China
handongfeng@gmail.com
http://handongfeng.googlepages.com/home
Zongcheng Li
School of Engineering Technology,
Shandong University of Technology, Zibo, P. R. China

Abstract

This paper presents a new interest local regions descriptors method based on Hilbert-Huang Transform. The neighborhood of the interest local region is decomposed adaptively into oscillatory components called intrinsic mode functions (IMFs). Then the Hilbert transform is applied to each component and get the phase and amplitude information. The proposed descriptors samples the phase angles information and amalgamates them into 10 overlap squares with 8-bin orientation histograms. The experiments show that the proposed descriptors are better than SIFT and other standard descriptors. Essentially, the Hilbert-Huang Transform based descriptors can belong to the class of phase-based descriptors. So it can provides a better way to overcome the illumination changes. Additionally, the Hilbert-Huang transform is a new tool for analyzing signals and the proposed descriptors is a new attempt to the Hilbert-Huang transform.

1 Introduction

Efficient local region descriptors are very useful in computer vision applications such as matching, indexing, retrieval and recognition. Commonly, the procedure of correspondences problem is not complex and includes several stages as shown in Fig.1: (1) detect the interest local regions; (2) normalize the local regions size and the main orientation; (3) compute their descriptors; (4) match the local regions using certain similar matching methods.

The object of local region detection algorithm is to specify the locations and the scales of the features. The detectors should be invariant to translation, rotation, scale, affine transform. Many scale/affine local regions detectors are proposed in the past few years [1, 2, 3, 4, 5, 6, 7]. Given invariant region detectors, the remaining question is which are the

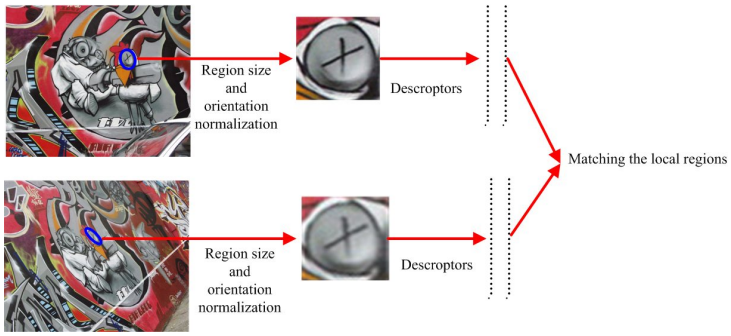


Figure 1: The main steps for the correspondences problem.

most appropriate descriptors to characterize the local regions. Many different methods for describing local interest regions have been developed [9, 10, 11, 12, 13, 14, 15, 3, 16, 8].

In this paper, a new signal process tool Hilbert-Huang transform is used to describe the local regions. The local regions are decomposed by bidimensional empirical mode decomposition (BEMD) and several intrinsic mode functions (IMFs) and the residual part are obtained. Then, Hilbert spectral analysis are conducted to describe the local regions. The Hilbert-Huang transform is locally adaptive and suitable for analysis of non-linear or non-stationary signals. Because the local regions we would process always occurs in the non-stationary regions, it would be very suitable to describe the local regions using Hilbert-Huang transform. At the same time, it shares the good features with wavelet and Fourier analyses. Also, studies in [17] shows that it can provide much better temporal and frequency resolutions than wavelet and Fourier analyses.

The experiments on standard data set show that the proposed descriptors have better results for the image illumination changes and geometry transforms. Additionally, the proposed algorithm is a new attempt to the Hilbert-Huang transform.

The rest of this paper is organized as follows. In section 2, we discuss some issues about Hilbert-Huang transform. In section 3, the Hilbert-Huang transform based local regions descriptors are described in detail. In section 4, the experiments and comparisons are given. The results of the real image experiments are demonstrated. The empirical comparisons among our approach and other five standard methods are made. The conclusions and future work are discussed in section 5.

2 Hilbert-Huang Transform

Recently, Huang et al. [17] have introduced the empirical mode decomposition (EMD) method for analyzing data. Empirical mode decomposition (EMD) is a general nonlinear, non-stationary signal processing method. So it is very suitable for describing local structure or local texture. EMD has been used in many fields such as in [18, 19, 20]. The major advantage of the EMD is that the basis functions are derived from the signal itself. Hence, the analysis is adaptive, in contrast to the wavelet method where the basis functions are fixed. The central idea of EMD is to decompose a time series into a finite and often small number of intrinsic mode functions (IMFs). As discussed in [17], an intrinsic

mode function (IMF) should satisfy two conditions: (1) in the whole data set, the number of extrema and the number of zero crossings must either equal or differ at most by one; (2) at any point, the mean value of the envelope defined by the local maxima and the envelope defined by the local minima is zero. It is emphasized by Huang that the second condition is very important to EMD, especially for non-stationary signal (such as interest local regions). Paper [17] has a more insight on EMD.

Given a signal $x(t)$, the EMD can be represented as:

$$x(t) = \sum_{i=1}^{N} IMFs_i(t) + r(t), \tag{1}$$

where N is the number of IMFs and $r(t)$ is the residue of the signal.

Hilbert-Huang transform has the advantages in analysing the local structure. Hilbert-Huang transform includes two parts: 1) EMD; 2) Hilbert spectrum analysis (HSA). After doing EMD on the signal, some IMFs can be obtained. The next step is to analyse the $IMFs$ using HSA. The main idea of HSA is to construct complex signal for the analyzed real signal. The positive part of the spectrum of the initial real signal $x(t)$ is multiplied by two and the negative part is set to zero. Such spectrum corresponds to complex signal $z(t)$ whose imaginary part is equal to the Hilbert transform of the real part that equals to $x(t)$ as equations (2)and (3).

$$z(t) = x(t) + iy(t), \tag{2}$$

$$y(t) = H(x(t)) = v.p. \int_{-\infty}^{+\infty} \frac{x(\tau)}{\pi(t-\tau)} d\tau, \tag{3}$$

where p indicates the Cauchy principal value.

Though the above definitions are defined for any function $x(t)$ which satisfies existence conditions for the integral (3), the physical meaning of parameters phase and amplitude information is obvious only if $x(t)$ belongs to the class of monocomponent functions, i.e. the number of its extremes and the number of zero-crossings differ at most by 1 and the mean between the upper and lower envelopes equals to zero. In practice, most of the image signals are not monocomponent. By the definition of EMD in section2, the conditions are satisfied after empirical mode decomposition. EMD is very suitable for the Hilbert (or Riesz in multidimensional case) analysis.

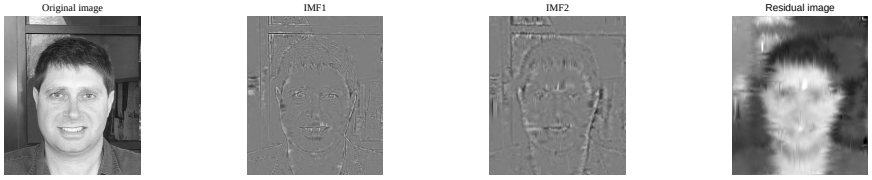
For bidimensional signals, a similar algorithm called BEMD can be used to analysis the image signal. The principle is similar with EMD. The BEMD procedure is shown in Table1. In Fig.2, a two-level BEMD performed on a face image is shown. From Fig.2, it is clear that IMF_1 and IMF_2 have much useful information. The BEMD indeed provides the multi-scale representation of the local regions. It is shown [21] that the extracted local features have direct semantic interpretation. It contains the pattern structures from the finest to the coarsest. So the descriptors can be extracted by Hilbert analysis from IMF_1, IMF_2 and residue part.

Image is two dimensional signal and the Riesz transform [21] is a multidimensional generalization of the Hilbert transform. Because $I(x,y) = \sum_{i=1}^{n} IMF_i(x,y) + I_{r_n}(x,y)$, two dimensional complex signal can be expressed as,

$$IMF_{tA}(x,y) = IMF_t(x,y) + iIMF_{tH}(x,y), t = 1,2,...n, \tag{4}$$

Table 1: The BEMD algorithm

BEMD Algorithm:
(1) Initialization, $I_{r_0}(x,y) = I(x,y), i = 1$;
(2) Extracting ith $IMFs$:
1) Let $h_0 = I_{r_i}(x,y), k = 1$;
2) For $h_{k+1}(k)$, extracting local maxima $(m_{max,k-1})$ and minima $(m_{min,k-1})$;
3) Creating upper and lower envelope by spline interpolation of the local maxima $(m_{max,k-1})$ and minima $(m_{min,k-1})$;
4) Computing mean value of the envelope $m_{k-1}(x,y)$;
$m_{mean,k-1}(x,y) = \frac{1}{2}(m_{max,k-1}(x,y) + m_{min,k-1}(x,y))$;
5) Computing $h_{mean,k}(x,y) = h_{k-1}(x,y) - m_{k-1}(x,y)$;
6) Checking if mean signal is close enough to zero. Yes: $h_k(x,y) = IMF_i(x,y)$, otherwise go to 2), and set $k = k+1$;
(3) $I_{r_i}(x,y) = I_{r_{i-1}}(x,y) - IMF_i(x,y)$;
(4) If the $extreme > 2$ in $I_{r_i}(x,y)$, then go to (2), and set $i = i+1$, otherwise finish. The final result:
$I(x,y) = \sum_{i=1}^{n} IMF_i(x,y) + I_{r_n}(x,y)$;



(a) (b) (c) (d)

Figure 2: An example of BEMD. (a) The original image. (b) IMF_1 after BEMD. (d) IMF_2 after BEMD. (b) The residual image after BEMD.

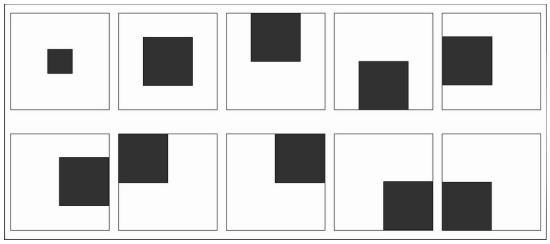


Figure 3: The illustration of overlap division for a normalized region. The local region is divided into 10 overlap regions. For each square the descriptors are computed. The formulation of division method is different from SIFT and other non-overlap methods.

where $IMF_t(x,y)$ and $IMF_{tH}(x,y)$ are called the real and complex parts of $I_{tA}(x,y)$.

So we can get the amplitude and the phase as equations (5) and (6).

$$A(x,y) = \sqrt{IMF_t^2(x,y) + IMF_{tH}^2(x,y)}, \tag{5}$$

$$\theta(x,y) = \arctan \frac{IMF_{tH}(x,y)}{IMF_t(x,y)}. \tag{6}$$

It can be denoted in a compact expression,

$$G(x,y) = A(x,y).exp(i\theta(x,y)). \tag{7}$$

In order not to lose information, the residual part as equation(8) can be analyzed by the same way.

$$I_{r_nA}(x,y) = I_{r_n}(x,y) + iI_{r_{nH}}(x,y). \tag{8}$$

Commonly, nature images always have complex texture structure and the local regions we are interested always occur in these regions. Similar with $1D$ signal, image signal is suitably analyzed by Hilbert-Huang transform.

3 Hilbert-Huang Transform Based Local Region Descriptors

Though the phase contains more information than amplitude, the phase is sensitive to the image transforms. In order to overcome this disadvantage, we project the phase onto eight orientations for each pixel as in SIFT [3].

DOG detector is used to detect local regions over scales. The characteristic scale determines the structure of the local region. The local region is rotated to its main orientation and then is normalized to 41×41. The local region is performed by two levels BEMD which can describe the detail local structure enough. Together with the residual image, the proposed algorithm will get 3 "images"(IMF_1, IMF_2, and I_{rn}).

In order to reduce the feature dimensions and maintain the spatial information , the local region is spatially divided into 10 overlap square regions. This division method can

(a) (b) (c) (d)

Figure 4: Some test images. (a) graf image.(b) boat image. (c) car image. (d) motorcycle image.

get better result than the non-overlap in [3]. The division scheme can provide high overlap ratio with enough descriptors dimensions. In Fig.3, a division example is illustrated. In each square region, the phase angle $\theta(x,y)$)is projected onto eight orientations with the amplitude $A(x,y)$ at each position. In order to restrain the sensitivity to illumination changes, the amplitude is controlled using equation(9) as in [22]. By this way, the saturated amplitude is roughly constant for large amplitude. With the eight orientations, 10 squares and 3 images(IMF_1, IMF_2, and I_{rn}), the total dimensions of the descriptors are 240.

$$\widetilde{A}(x,y) = 1 - \exp\frac{-A^2(x,y)}{2}. \tag{9}$$

In practice, it is not necessary to conduct BEMD for every local region. In fact, the BEMD can be performed on the whole image at the beginning of the algorithm. This tip can speed up the running time.

4 Experiments and Comparisons

4.1 Matching Strategy and Comparison Criterion

Matching method is important to the final performance. For different problems, different matching methods should be used. Nearest neighbour matching and ratio matching are often used.

(1) Nearest neighbour matching: A and B are matched if the descriptor D_B is the nearest neighbor to D_A and if the distance between them is below a threshold. With this approach a descriptor has only one match.

(2) Ratio matching: this method is similar to nearest neighbor matching except that the thresholding is applied to the distance ratio between the first and the second nearest neighbour. Thus the regions are matched if $\|D_A - D_B\|/\|D_A - D_C\| < t$ where D_B is the first and D_C is the second nearest neighbour to D_A.

In this paper the ration matching method is used. We use *recall* and *precision* as the performance evaluation metric. They are defined as, $recall = \frac{\#correctmatches}{\#correspondences}$ and $precision = \frac{\#correctmatches}{\#correctmatches+\#false}$.

Two points v_i and v_j are a pair of correct match if the error in relative location is less than 3 pixels, which means v_i and v_j should satisfy $L_{v_i} - HL_{v_j} < 3$, Where H is the homography between v_i and v_j.

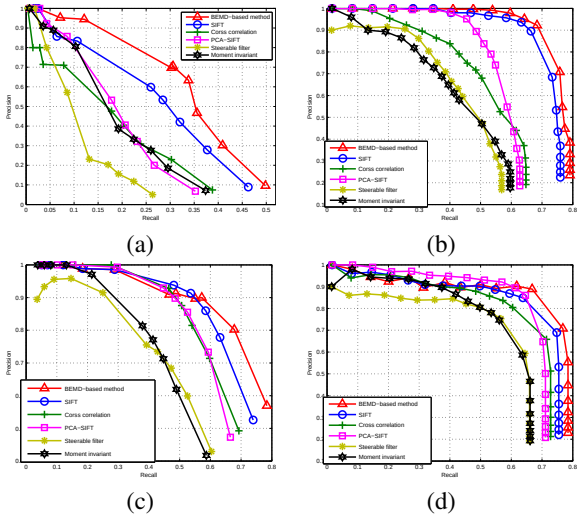


Figure 5: Performances evaluation for several image changes. (a)ROC curve under affine changes for graf image. (b) ROC curve under scale + rotation changes for boat image. (c) ROC curve under illumination changes for car image. (d) ROC curve under blur changes for motorcycle image.

4.2 Experiments Comparisons

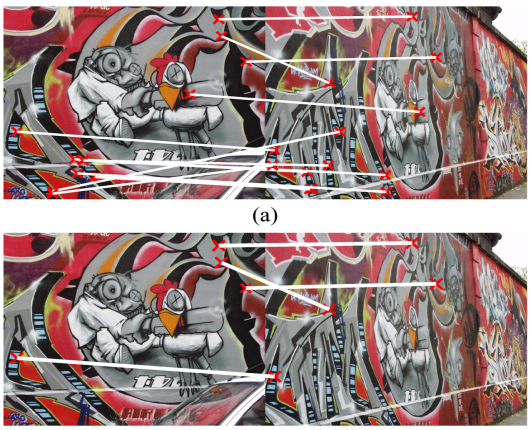
The data set is available from VGG Lab [23]. Some of the images are shown in Fig.4. We compare six different descriptors which include: 1) the proposed algorithm; 2) SIFT [3]; 3) cross correlation; 4) PCA-SIFT [16]; 5) steerable filters [11]; 6) moment invariant [9]. We generate the *recall* vs *precision* graph by changing the threshold for six different descriptors.

In Fig.5, the ROC curves are demonstrated for different image changes which include affine changes(graf image), scale+rotation(boat image), illumination changes(car image) and blur changes(motorcycle image). In our experiments, the proposed descriptors show its superior for illumination changes and affine transform. The reason is that the proposed descriptors is a phased-based descriptors essentially. For the illumination changes, the phase information (corresponding to the orientation) is more robust than other methods.

In Fig.6, a matching example is given for view point change. We compare the proposed descriptors with SIFT. It can be seen from the example that the proposed algorithm can match more accurately than SIFT.

5 Conclusions

In this paper, a new local region descriptors is presented. The main contribute of this paper is introducing Hilbert-Huang transform to local descriptors. We elaborately design an efficient compact local descriptors based on the BEMD and Hilbert transform. The main idea of the proposed descriptors is to analyze the IMFs by Hilbert spectrum. The corre-



(a)

(b)

Figure 6: Matching comparison proposed method with SIFT. The mismatches are drawn in white lines with red cross. (a) The mismatches obtained by standard SIFT. SIFT: 13/50 mismatches. (b) The mismatches obtained by the proposed algorithm. Proposed Method: 4/50 mismatches.

sponding experiments are promising. The new descriptors show its advantages especially for illumination changes and common geometry transforms.

References

[1] T. Lindeberg.: Feature detection with automatic scale selection. International Journal of Computer Vision, Vol. 30, 1998), No. 2, pp. 79–116.

[2] C. S. K. Mikolajczyk.: Indexing based on scale invariant interest points. 8th International Conference on Computer Vision, Institute of Electrical and Electronics Engineers Inc, 2001, pp. 525–531.

[3] D. G. Lowe.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, Vol. 60, 2004, No. 2, pp. 91–110.

[4] T. Tuytelaars. and L. V. Gool.: Wide baseline stereo matching based on local, affinely invariant regions. The Eleventh British Machine Vision Conference, 2000, pp. 412–425.

[5] T. Tuytelaars and L. Van Gool.: Matching widely separated views based on affine invariant regions. International Journal of Computer Vision, Vol. 59, 2004, No. 1, pp. 61–85.

[6] O. C. J. Matas, M. Urban and T. Pajdla.: Robust wide baseline stereo from maximally stable extremal regions. 13th British Machine Vision Conference, 2002, pp. 384–393.

[7] K. Mikolajczyk and C. Schmid.: An affine invariant interest point detector. Computer Vison - Eccv 2002, 2002, pp. 128–142.

[8] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir and L. van Gool.: A comparison of affine region detectors. International Journal of Computer Vision, Vol. 65, 2005, No. 1-2, pp. 43–72.

[9] L. J. V. Gool, T. Moons and D. Ungureanu.: Affine/ photometric invariants for planar intensity patterns. 4th European Conference on Computer Vision, Vol. 1, Springer-Verlag, 1996, pp. 642-651.

[10] F. Mindru, T. Tuytelaars, L. Van Gool and T. Moons.: Moment invariants for recognition under changing viewpoint and illumination. Computer Vision and Image Understanding, 2004, No. 1-3, pp. -3-27.

[11] W. T. Freeman and E. H. Adelson.: The design and use of steerable filters. Ieee Transactions on Pattern Analysis and Machine Intelligence, Vol. 13, 1991, No. 9, pp. 891–906.

[12] B. t. H. R. L.M.T. Florack, J.J Koenderink, and M.A. Viergever.: General intensity transformations and differential invariants. Journal of Mathematical Imaging and Vision, Vol. 4, 1994, No. 2, pp. 171–187.

[13] A. Baumberg.: Reliable feature matching across widely separated views. CVPR 2000: IEEE Conference on Computer Vision and Pattern Recognition, Institute of Electrical and Electronics Engineers Computer Society, Los Alamitos, CA, USA, 2000, pp. 774–781.

[14] F. Schaffalitzky and A. Zisserman.: Multi-view matching for unordered image sets, or How do i organize my holiday snaps? Computer Vison - Eccv 2002, 2002, pp. 414–431.

[15] G. Carneiro and A. D. Jepson.: Multi-scale phase-based local features. 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Institute of Electrical and Electronics Engineers Computer Society, 2003, pp. I/736–I/743.

[16] Y. Ke and R. Sukthankar.: Pca-sift: A more distinctive representation for local image descriptors. Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004, Institute of Electrical and Electronics Engineers Computer Society, Piscataway, NJ 08855-1331, United States, 2004, pp. II506-II513.

[17] N. E. Huang, Z. Shen, S. R. Long, M. L. C. Wu, H. H. Shih, Q. N. Zheng, N. C. Yen, C. C. Tung and H. H. Liu.: The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. Proceedings of the Royal Society of London Series a-Mathematical Physical and Engineering Sciences, Vol. 454, 1998, No. 1971, pp. 903–995.

[18] Z. Liu, H. Wang and S. Peng.: Texture segmentation using directional empirical mode decomposition. 2004 International Conference on Image Processing, ICIP

2004, Institute of Electrical and Electronics Engineers Computer Society, Piscataway, NJ 08855-1331, United States 2004, pp. 279–282.

[19] P. Flandrin, G. Rilling and P. Goncalves.: Empirical mode decomposition as a filter bank. Ieee Signal Processing Letters, Vol. 11, 2004, No. 2, pp. 112–114.

[20] G. Rilling, P. Flandrin and P. Goncalves.: Empirical mode decomposition, fractional gaussian noise and hurst exponent estimation. 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '05, Institute of Electrical and Electronics Engineers Inc., Piscataway, NJ 08855-1331, United States, 2005, pp. IV489–IV492.

[21] J. C. Nunes, Y. Bouaoune, E. Delechelle. Texture analysis based on local analysis of the Bidimensional Empirical Mode Decomposition. Machine Vision and Application, 2005, No. 16, pp, 177–188.

[22] G. Carneiro and A. Jepson. Phase-based local features. In European Conference on Computer Vision, Copenhagen, Denmark, May 2002.

[23] http://www.robots.ox.ac.uk/∼vgg/research/affine/

Incremental LDA Learning by Combining Reconstructive and Discriminative Approaches *

Martina Uray[1], Danijel Skočaj[2], Peter M. Roth[1],
Horst Bischof[1] and Aleš Leonardis[2]

[1] Institute for Computer Graphics and Vision, Graz University of Technology
{uray, pmroth, bischof}@icg.tugraz.at

[2] Faculty of Computer and Information Science, University of Ljubljana
{danijel.skocaj, ales.leonardis}@fri.uni-lj.si

Abstract

Incremental subspace methods have proven to enable efficient training if large amounts of training data have to be processed or if not all data is available in advance. In this paper we focus on incremental LDA learning which provides good classification results while it assures a compact data representation. In contrast to existing incremental LDA methods we additionally consider reconstructive information when incrementally building the LDA subspace. Hence, we get a more flexible representation that is capable to adapt to new data. Moreover, this allows to add new instances to existing classes as well as to add new classes. The experimental results show that the proposed approach outperforms other incremental LDA methods even approaching classification results obtained by batch learning.

1 Introduction

When discussing object classifiers one faces two main philosophies, generative and discriminative classifiers. Whereas generative models try to find an efficient representation of the data, discriminant classifiers are particularly designed for classification tasks. Compared to generative methods this allows to train more specific classifiers resulting in higher recognition rates. In fact, several studies (e.g., [1, 13]) have shown that discriminative classifiers outperform generative models (if enough training data is available).

But discriminative classifiers have several drawbacks, e.g., a huge amount of training data is necessary, they are not robust (neither during learning nor during evaluation), or they can not take advantage of unlabeled data. Hence, there was a considerable interest to bring both philosophies together (e.g., [2, 7, 9–12]). Most of these approaches are based on two-stages (e.g., [7, 9, 10]). In the first stage a generative model (probabilistic constellation model [7], clustering algorithm [9], probabilistic PCA [10]) is estimated

*This research has been supported by: Research program Computer Vision P2-0214 (Slovenian Ministry of Higher Education, Science and Technology), EU FP6-004250-IP project CoSy, EU FP6-511051 project MOBVIS, EU FP6-507752 NoE MUSCLE IST, Austrian Joint Research Project Cognitive Vision under projects S9103-N04 and S9104-N04 and FFG project AUTOVISTA 813395 under the FIT-IT program.

and in the second stage a discriminant classifier (SVM on Fisher Scores [7], multi-layer perceptron [9], negative samples [10]) is built from the generative model.

In the field of subspace methods this combination is mainly addressed by classifier fusion of a generative model (mostly PCA) and a discriminative model (e.g., LDA) which improves the recognition results (e.g., [11, 12]). Similarly, the combination of PCA and LDA was addressed by Fidler et al. [2]. They constructed a new basis that contains all discriminative information but in addition also contains sufficient reconstructive information to enable a robust classification. Due to its reconstructive properties we will further refer PCA to as a reconstructive model.

But none of these approaches consider the benefits of combining generative and discriminative methods for incremental learning which is desirable for two main reasons. First, if not all data is given in advance (e.g., when processing a video stream) a batch method can not be applied. Second, incremental learning requires less memory and has lower computational costs. In the following we will focus on incremental LDA learning. Existing incremental LDA methods, e.g., [6, 15, 18] are based on directly updating the between class scatter matrix and the within class scatter matrix. That is, they consider mostly discriminative information focusing on specific features that best discriminate the data seen so far. To illustrate that using both, discriminative and reconstructive information, is beneficial consider the following example. Given an LDA classifier trained from a small number of samples to separate three classes. As can be seen in Figure 1 this *initial classifier* separates the classes well in the projected LDA space. This subspace representation is then updated with new instances in two different ways, using discriminative information only and using discriminative as well as reconstructive information. From Figure 1 it can be seen that for the first case (*LDA*) the representation gets more and more dispersed since the classifier is biased by the starting model and can not adapt to the new information. Conversely, if discriminative and reconstructive information is used (*PCA & LDA*) the representation remains compact and still separates the three clusters well.

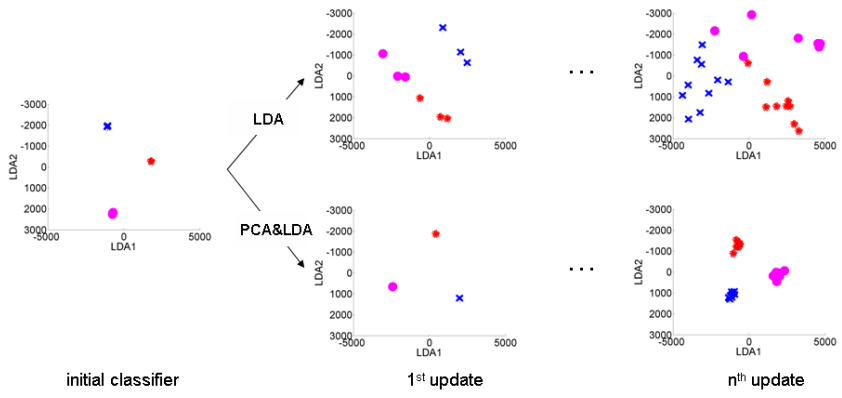


Figure 1: Incremental discriminative subspace learning: using only discriminative information the incrementally estimated model is biased by the first samples and the representation get dispersed (LDA); using reconstructive and discriminative information a more generic model is estimated which yields well separable clusters (PCA & LDA).

The main contribution of this paper is an incremental discriminative subspace learning algorithm that combines reconstructive and discriminative information during training. In particular, we use Linear Discriminant Analysis (LDA) [3] as discriminative model and Principal Component Analysis (PCA) [8] as reconstructive model. For LDA learning it is common to apply PCA as a preprocessing step for dimensionality reduction or to handle singularity problems (e.g., [1, 17]). Conversely, our approach implicitly combines both models. Since every update increases the size of the representation the updated subspace has to be truncated. Usually the least significant (reconstructive) directions are skipped but this may result in a loss of discriminative information. To avoid this we propose a more sophisticated truncation step. By applying an appropriate constructed orthogonal projection the updated subspace is projected onto another subspace of desired size. Thus, preserving the reconstructive as well as the discriminative information. Moreover, this update procedure allows to incrementally update the current representation by adding new instances to existing classes as well as by adding new classes.

The outline of the paper is as follows: In Section 2 we deduce the combined reconstructive/discriminative subspace projection and show how it can be applied for incremental learning. Next, in Section 3 we demonstrate the advantages of our approach compared to other incremental methods on small as well as large data sets and for different kinds of updates. Finally, we give a conclusion and an outlook in Section 4.

2 Incremental Updating by Subspace Combination

2.1 Subspace Methods

The goal of subspace methods is to find a projection that transforms the training data (images) such that new unknown images can be efficiently classified. For batch methods it is assumed that all training data is given in advance. Formally, given n images $\mathbf{x}_j \in \mathbb{R}^m$ represented as vectors which are organized in a matrix $\mathbf{X} = [\mathbf{x}_1, \ldots \mathbf{x}_n] \in \mathbb{R}^{m \times n}$. Let $\mu \in \mathbb{R}^m$ be the mean image and let c be the total number of classes. In the following we take a closer look at the subspace methods that are relevant for our approach.

Principal Component Analysis [8] is an unsupervised method that estimates a low dimensional representation of the data that minimizes the squared reconstruction error. Therefore, the data $\mathbf{X}$ is projected onto a lower-dimensional subspace by

$$\mathbf{A} = \mathbf{U}^T \left(\mathbf{X} - \mu \mathbf{1}_{1 \times n} \right) , \tag{1}$$

where the projection matrix $\mathbf{U}$ is built from the eigenvectors of the covariance matrix of $\mathbf{X}$ and $\mathbf{1}_{m \times n}$ denotes an $m \times n$ matrix of ones. Usually only k, $k \ll n \ll m$, columns are needed to reconstruct $\mathbf{X}$ to a desired accuracy:

$$\mathbf{X} \approx \mathbf{U}_k \mathbf{A}_k + \mu \mathbf{1}_{1 \times n} , \tag{2}$$

where $\mathbf{U}_k = [\mathbf{u}_1, \ldots, \mathbf{u}_k] \in \mathbb{R}^{m \times k}$ is the truncated reconstructive basis and the rows of $\mathbf{A}_k = [\mathbf{a}_1^T, \ldots, \mathbf{a}_n^T]^T \in \mathbb{R}^{k \times n}$ are the image representations in the truncated subspace.

Linear Discriminant Analysis [3], in contrast, is a supervised method that also integrates the class label information of the training samples. It seeks for $(c-1)$ hyperplanes that are capable to separate the given data. This is realized by a projection

$\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_{(c-1)}] \in \mathbb{R}^{m \times (c-1)}$ that minimizes the intra-class scatter while it maximizes the inter-class scatter. Hence, a test sample $\mathbf{x}$ is projected onto the subspace by

$$g(\mathbf{x}) = \mathbf{W}^T (\mathbf{x} - \mu) \; . \tag{3}$$

The class label l is then assigned according to the result of a nearest neighbor classification. Therefore, the Euclidean distances d of the projected sample $g(\mathbf{x})$ and the class centers v_i are compared:

$$l = \min_{i=1,\ldots,c} d(g(\mathbf{x}), v_i) \; . \tag{4}$$

2.2 Incremental LDA Learning

Incremental subspace methods allow to directly add new images to an already built lower-dimensional representation. Thus, the original images can be discarded directly after they have been processed. To avoid that thus discriminative information is lost in the following we will deduce an incremental update scheme for LDA that is based on the ideas of augmented subspaces [2] that were originally indented for robust classification (*ILDAaPCA*). Therefore, we first create an augmented PCA subspace by augmenting the k dimensional reconstructive subspace with additional $c-1$ vectors containing discriminative information. Those supplementary vectors are constructed from vectors that would be discarded when truncating the subspace to k-dimensions. In this way, the full discriminative information is maintained. Second, we build the actual LDA representation from the thus obtained augmented subspace.

Assuming that a subspace representation was already built from n images the current augmented PCA subspace is given by the *aPCA* vectors $\mathbf{U}^{(n)} \in \mathbb{R}^{m \times (k+c-1)}$, the *aPCA* coefficients $\mathbf{A}^{(n)} \in \mathbb{R}^{(k+c-1) \times n}$, and the mean vector $\mu^{(n)} \in \mathbb{R}^{m \times 1}$; the current augmented LDA representation is given by the vectors $\mathbf{V}^{(n)} \in \mathbb{R}^{(k+c-1) \times (c-1)}$ and the class centers $v_i^{(n)} \in \mathbb{R}^{(c-1) \times 1}$. To update the subspace representation with a new image $\mathbf{x}^{(n+1)}$ first the augmented subspace is updated using an incremental PCA method (*IPCA*) [16] which is outlined in Algorithm 1.

Second, the actual LDA step is carried out on the extended representation (Algorithm 2, step 1) giving the updated augmented LDA space $\mathbf{V}$ and the corresponding class centers $v_i^{(n+1)}$. Here, $\mathbf{U}$ and $\mathbf{V}$ represent those subspaces that correctly describe the updated information. That is, $g(\mathbf{x})$ projects onto a subspace describing the old data as well as the new input image. But as a consequence the dimension of the subspace representation is increased: $\mathbf{U} \in \mathbb{R}^{m \times (k+c)}$, $\mathbf{A} \in \mathbb{R}^{(k+c) \times (n+1)}$, $\mathbf{V} \in \mathbb{R}^{(k+c) \times (c-1)}$.

Finally, in order to retain a pre-specified size we have to truncate the subspaces. To avoid that discriminative information is lost we project these matrices onto another subspace having the desired size while preserving the full discriminative information. Therefore, we want to keep the first k vectors of the subspaces unchanged. This is achieved by splitting the LDA vectors into sub-matrices $\mathbf{V}_k \in \mathbb{R}^{k \times (c-1)}$ and $\mathbf{V}_c \in \mathbb{R}^{c \times (c-1)}$ (Algorithm 2, step 2). The orthogonalized matrix $\widetilde{\mathbf{V}}_c$ (Algorithm 2, step 3) has exactly the desired properties (preserving the discriminative information).

Algorithm 1 : IPCA on Augmented Basis

Input: Current augmented principal subspace: aPCA vectors $\mathbf{U}^{(n)}$, aPCA coefficients $\mathbf{A}^{(n)}$ and mean vector $\mu^{(n)}$; new input image $\mathbf{x}^{(n+1)}$

Output: New augmented principal subspace: aPCA vectors $\mathbf{U}$, aPCA coefficients $\mathbf{A}$ and mean vector $\mu^{(n+1)}$

1: Project $\mathbf{x}^{(n+1)}$ onto current subspace: $\mathbf{a} = \mathbf{U}^{(n)\top}(\mathbf{x}^{(n+1)} - \mu^{(n)})$
2: Reconstruct: $\mathbf{y} = \mathbf{U}^{(n)}\mathbf{a} + \mu^{(n)}$
3: Compute residual vector: $\mathbf{r} = \mathbf{x}^{(n+1)} - \mathbf{y}$
4: Append new basis vector $\mathbf{r}$: $\mathbf{U}' = \begin{bmatrix} \mathbf{U}^{(n)} & \frac{\mathbf{r}}{\|\mathbf{r}\|} \end{bmatrix}$
5: Determine coefficients in new basis: $\mathbf{A}' = \begin{bmatrix} \mathbf{A}^{(n)} & \mathbf{a} \\ \mathbf{0} & \|\mathbf{r}\| \end{bmatrix}$
6: Perform PCA on $\mathbf{A}'$ obtaining mean value μ'' and eigenvectors $\mathbf{U}''$
7: Project coefficient vectors to new basis: $\mathbf{A} = \mathbf{U}''^{\top}(\mathbf{A}' - \mu''\mathbf{1}_{1\times(n+1)})$
8: Rotate subspace $\mathbf{U}'$ for $\mathbf{U}''$: $\mathbf{U} = \mathbf{U}'\mathbf{U}''$
9: Update mean: $\mu^{(n+1)} = \mu^{(n)} + \mathbf{U}'\mu''$

Hence, we can construct a projection matrix $\mathbf{M} \in \mathbb{R}^{(k+c)\times(k+c-1)}$ containing an identity matrix $\mathbf{I} \in \mathbb{R}^{k\times k}$, the matrix $\widetilde{\mathbf{V}}_c \in \mathbb{R}^{c\times(c-1)}$, and the zero matrices $\mathbf{0}_c \in \mathbb{R}^{c\times k}$ and $\mathbf{0}_k \in \mathbb{R}^{k\times(c-1)}$ (see Algorithm 2, step 4). Finally, the current subspaces are projected onto $\mathbf{M}$ and we obtain $\mathbf{U}^{(n+1)} \in \mathbb{R}^{m\times(k+c-1)}$, $\mathbf{A}^{(n+1)} \in \mathbb{R}^{(k+c-1)\times(n+1)}$, and $\mathbf{V}^{(n+1)} \in \mathbb{R}^{(k+c-1)\times(c-1)}$ (Algorithm 2, step 5).

Algorithm 2 : LDA on Updated Augmented Basis

Input: Updated augmented principal subspace: aPCA vectors $\mathbf{U}$, aPCA coefficients $\mathbf{A}$

Output: New subspaces of pre-specified size: aPCA vectors $\mathbf{U}^{(n+1)}$, aPCA coefficients $\mathbf{A}^{(n+1)}$; LDA vectors $\mathbf{V}^{(n+1)}$ and class centers $v_i^{(n+1)}$

1: Perform LDA on $\mathbf{A}$ obtaining LDA vectors $\mathbf{V}$ and class centers $v_i^{(n+1)}$
2: Split $\mathbf{V} = \begin{bmatrix} \mathbf{V}_k \\ \mathbf{V}_c \end{bmatrix}$
3: Orthogonalize $\mathbf{V}_c$: $\widetilde{\mathbf{V}}_c = \mathbf{V}_c(\mathbf{V}_c^T\mathbf{V}_c)^{-1/2}$
4: Build projection matrix $\mathbf{M}$: $\mathbf{M} = \begin{bmatrix} \mathbf{I}_k & \mathbf{0}_k \\ \mathbf{0}_c & \widetilde{\mathbf{V}}_c \end{bmatrix}$
5: Project $\mathbf{U}$, $\mathbf{A}$ and $\mathbf{V}$: $\mathbf{U}^{(n+1)} = \mathbf{U}\mathbf{M}$, $\mathbf{A}^{(n+1)} = \mathbf{M}^{\top}\mathbf{A}$, $\mathbf{V}^{(n+1)} = \mathbf{M}^{\top}\mathbf{V}$

The main point is that by this truncation step the projection $g(\mathbf{x})$ does not change:

$$\begin{aligned} g^{(n+1)}(\mathbf{x}) &= \left(\mathbf{U}^{(n+1)}\mathbf{V}^{(n+1)}\right)^{\top}(\mathbf{x} - \mu^{(n+1)}) &= \\ &= \left(\mathbf{U}\mathbf{M}\mathbf{M}^{\top}\mathbf{V}\right)^{\top}(\mathbf{x} - \mu^{(n+1)}) &= \\ &= (\mathbf{U}\mathbf{V})^{\top}(\mathbf{x} - \mu^{(n+1)}) &= g(\mathbf{x}) \ . \end{aligned} \tag{5}$$

Thus, while still retaining the dimensions of the subspaces no discriminative information is lost.

3 Experimental Results

To show the benefits of the proposed incremental discriminative subspace method we performed two experiments that are also relevant for practical applications: adding new instances to existing classes (e.g., adding a new view of an already known object) and adding new classes (e.g., adding a new person to an authentication system). We compared the recognition results of the proposed *ILDAaPCA* method to the batch method (*batchLDA*) and two different incremental LDA approaches (*ILDAonK* and *ILDAonL*). *batchLDA* builds a new model from scratch in each update step using the same number of images as the incremental algorithms. *ILDAonK* updates a PCA basis truncated to the size $\widehat{k} = k + c - 1$ (to enable a comparison of subspaces of the same size[1]) and then discards the spare eigenvector in each update step. Thus, only reconstructive information is used. Conversely, *ILDAonL* updates the $(c-1)$-dimensional LDA basis directly; only discriminative information is used.

All experiments were performed on three datasets different in size, design, and topic. Color images were converted to gray scale and every second image of each class was used for training and the remaining ones for testing:

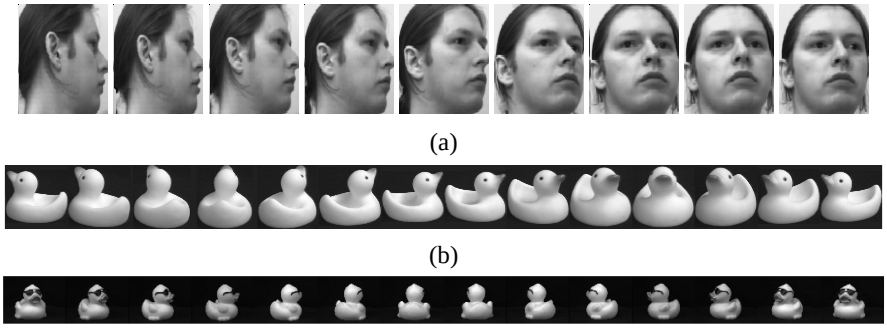
1. The *pre-cropped Sheffield Face Database* [5] (denoted as SFD) consists of 20 persons with at least 19 images of each individual covering poses from profile to frontal views (e.g., see Figure 2(a)). To have identical class sizes we used exactly 19 images per individual.

2. The *Columbia Image Database Library* [14] (COIL-100) consists of 100 objects with 72 colored images of views from 0 to 360 degrees in 5 degree steps (e.g., see Figure 2(b)).

3. The *Amsterdam Library Of Images* ALOI [4] is a color image collection of 1000 small objects, where we used the *Object Viewpoint Collection* (in their quarter resolution) which is similar to COIL-100. In contrast to COIL-100, where the objects are cropped to fill the full image in ALOI the images contain the objects in their original size as well as some background (e.g., see Figure 2(c)). For calculation purposes (to be able to compare our results to *batchLDA*) we used only the first 100 objects (denoted as ALOI-100).

Adding new instances: For the first experiment we assumed that all classes are present in the starting model. We took two images of each class, having 40 images in total for SFD and 200 images in total for ALOI-100 and COIL-100. Subsequently, in each iteration one additional training image is attached to all classes. For *batchLDA* the model was built from scratch with growing instance number for each class. Figure 3 shows the recognition results for SFD (Figure 3(a)) and ALOI-100 (Figure 3(b)) after each update step. This demonstrates that new images bring additional knowledge into the model improving the current representations resulting in an better performance of the classifier. Even though there are only minor differences between the compared methods for small datasets, our method can handle large data sets considerably better than the other incremental methods.

[1] The parameter k was chosen such that it encompasses 80% of the energy (defined as a fraction of the total variance) of the starting model unchanged during the entire experiment.



(a) (b) (c)

Figure 2: Sample images of one class from (a) SFD, (b) COIL-100, and (c) ALOI.

Moreover, the recognition rates of the proposed method are approaching those obtained by the batch method.

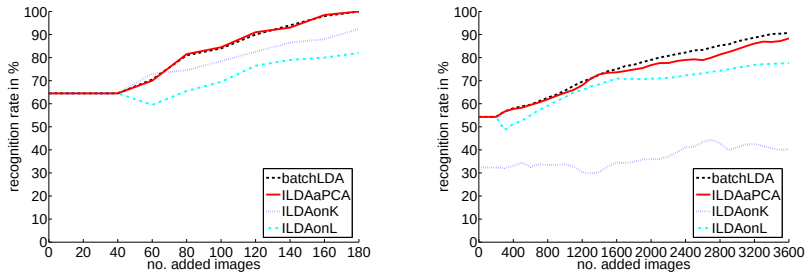


Figure 3: Recognition rates when adding new class instances: (a) SFD and (b) ALOI-100.

In addition, we demonstrate the importance of the proposed subspace truncation step (Algorithm 2, steps 2–5). Therefore, we compared the recognition rate on the Coil-100 database for our approach (see Figure 4(a)) and for an incremental method (denoted as *ILDAaPCAonK*) that starts with an augmented basis but simply truncates the last eigenvector in each update step. In Figure 4(b) it can be seen that the results are similar to *ILDAonK* showing the loss of important discriminative information.

Adding new classes: For the second experiment we assumed that only two classes (of full training size) are known in advance. Thus, initially we had a training base of size 18 for SFD and of size 72 for COIL-100 and ALOI-100. Subsequently, we added full classes. *batchLDA* was calculated in each step with full training size of actually known classes. Since the LDA subspace size depends on the number of classes it has to grow for each added class. *ILDAaPCA* handles this automatically; for the other methods we skipped the truncation step for the first image of each new class. For evaluation only test images from already known classes were used. The obtained recognition rates for SFD and ALOI-100

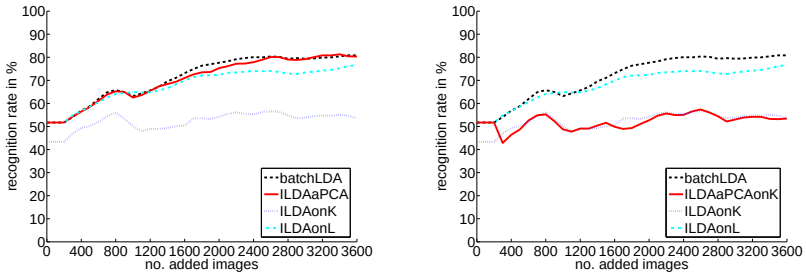


Figure 4: Update of the Coil-100 aPCA subspace: (a) $\mathbf{U}^{(n+1)} = \mathbf{UM}$; (b) $\mathbf{U}^{(n+1)} = \mathbf{U}_{k+c-1}$.

are shown in Figure 5(a) and Figure 5(b), respectively. It can be seen that due to the increasing complexity of the task (larger number of classes) the recognition rate drops for all methods even for *batchLDA*. But *ILDAaPCA* outperforms the other incremental methods.

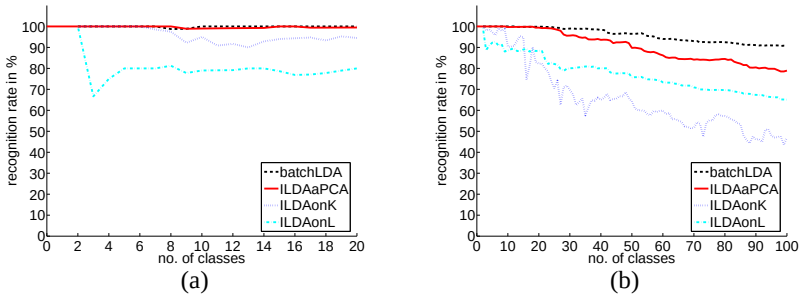


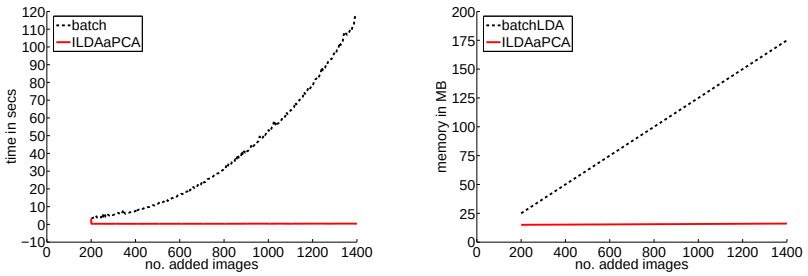
Figure 5: Recognition rates when adding new classes: (a) SFD and (b) ALOI-100.

Since for batch learning all training images are given and the basis vectors are not truncated the full information is available. Thus, it finds the most suitable hyperplanes and *batchLDA* always yields the best recognition rates. Updating the low-dimensional discriminative representations directly (*ILDAonL*) is not capable to adopt to a new task since it focuses only on initial given discriminative features. Conversely, by updating only the reconstructive representation (*ILDAonK*) too much discriminative information is lost. But the combination of both which is realized by *ILDAaPCA* is well suited for the task of incremental learning. Hence, the classification results are close to those of the batch method while *ILDAonL* and *ILDAonK* yield inferior classification results.

Time and Memory: As shown previously the recognition rate of the proposed incremental update scheme is comparable to batch learning. Thus, we can estimate an appropriate subspace even if not all images are given in advance. But incremental methods are more efficient which is shown in the following.

First, we compare the computational costs of *batchLDA* and *ILDAaPCA* . Figure 6(a) shows the computation time[2] needed for the update from n to $n+1$ images (evaluated for COIL-100). Since *ILDAaPCA* encompasses a model of fixed size only the representation matrix is growing. Thus, the computational costs for the updates are small. On the other hand, for *batchLDA* the representation has to be built from scratch for each additional image. Thus, the computation time is obviously increasing with each additional image.

Regarding memory requirements *ILDAaPCA* has to store only $\mathbf{U}, \mathbf{A}, \mathbf{V}$, and μ while *batchLDA* needs to store the full data matrix $\mathbf{X}$. Figure 6(b) illustrates the allocated memory in MB. It can be seen, that the memory requirements for *ILDAaPCA* are only slightly increasing since only the size of the lower-dimensional representations is growing. For *batchLDA* considerable more memory is allocated if new images are added. As a result for larger data sets batch computation is not possible while the incremental update always is applicable.



(a) (b)

Figure 6: Requirements for adding images: (a) time and (b) memory.

4 Conclusion

We proposed a method capable to incrementally update an existing LDA subspace. This is achieved by additionally incorporating reconstructive information to the discriminative model. The reconstructive property ensures an adequate representation of the training images and allows the discriminative information to focus on new features whenever it is necessary. Hence, our method is capable to add new instances of already known classes as well as adding unknown classes. The experiments have shown that our method outperforms other tested incremental subspace methods that keep only discriminative or reconstructive information. Thus, these methods are not able to adapt to new data, especially when processing large datasets. Furthermore, our method yields similar results as the batch approach but it is more efficient considering the computational costs and memory requirements. Further work will include the evaluation of the recognition rate if the influence of the reconstructive basis is growing with the number of represented classes. In addition, since the reconstructive representation also allows detection of outliers the proposed method can easily be extended in a robust way.

[2]These results were obtained on a PC with a 2xDual-Core Intel Xeon 5160, 3.0 GHz CPU and 16GB RAM.

References

[1] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. *PAMI*, 19(7):711–720, 1997.

[2] S. Fidler, D. Skočaj, and A. Leonardis. Combining Reconstructive and Discriminative Subspace Methods for Robust Classification and Regression by Subsampling. *PAMI*, 28(3):337–350, 2006.

[3] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

[4] J. M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders. The Amsterdam Library of Object Images. *Computer Vision*, 61(1):103–112, 2005.

[5] D. B. Graham and N. M. Allinson. *Face recognition: From theory to applications*, volume 163 of *NATO ASI Series F, Computer and Systems Sciences*, chapter Characterizing virtual eigensignatures for general purpose face recognition, pages 446–456. Springer, 1998.

[6] K. Hiraoka, S. Yoshizawa, K. Hidai, M. Hamahira, H. Mizoguchi, and T. Mishima. Convergence analysis of online linear discriminant analysis. In *Proc. of the Int. Joint Conf. on Neural Networks*, volume 3, pages 387–391, 2000.

[7] Alex D. Holub, Max Welling, and Pietro Perona. Combining generative models and fisher kernels for object recognition. In *Proc. IEEE Intern. Conf. on Computer Vision*, volume I, pages 136–143, 2005.

[8] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Educational Psychology*, 24:417–441, 1933.

[9] Yi Li, Linda G. Shapiro, and Jeff A. Bilmes. A generative/discriminative learning algorithm for image classifications. In *Proc. IEEE Intern. Conf. on Computer Vision*, volume II, pages 1605–1612, 2005.

[10] Ruei-Sung Lin, David Ross, Jongwoo Lim, and Ming-Hsuan Yang. Adaptive discriminative generative model and its applications. In *Proc. Conf. on Neural Information Processing Systems*, pages 801–808, 2005.

[11] X. Lu, Y. Wang, and A.K. Jain. Combining classifiers for face recognition. In *Proc. of the Int. Conf. on Multimedia and Expo*, volume 3, pages 13–16, 2003.

[12] G. L. Marcialis and F. Roli. Fusion of PCA and LDA for face verification. In *Proc. of of Post-ECCV Workshop on Biometric Authentication*, pages 30–37, 2002.

[13] A. M. Martínez and A. C. Kak. PCA versus LDA. *PAMI*, 23(2):228–233, 2001.

[14] S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (COIL-100). Technical Report CUCS-006-96, Department of Computer Science, Columbia University, 1996.

[15] S. Pang, S. Ozawa, and N. Kasabov. Incremental linear discriminassignmentant analysis for classification of data streams. *Systems, Man and Cybernetics, Part B*, 35(5):905–914, 2005.

[16] D. Skočaj and A. Leonardis. Weighted and robust incremental method for subspace learning. In *ICCV 2003*, volume 2, pages 1494–1501, 2003.

[17] J. Yang and J.-Y. Yang. Why can LDA be performed in PCA transformed space? *PR*, 36:563–566, 2003.

[18] J. Ye, H. Xiong Q. Li, H. Park, R. Janardan, and V. Kumar. IDR/QR: An Incremental Dimension Reduction Algorithm via QR Decomposition. In *Proc. of the ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 364–373, 2004.

18th British Machine Vision Conference (BMVC 2007), 10–13 September 2007, Coventry, U.K.

High Accuracy Computation of Rank-Constrained Fundamental Matrix

Yasuyuki Sugaya
Department of Information and Computer Sciences,
Toyohashi University of Technology, Toyohashi, Aichi 441-8580 Japan
sugaya@iim.ics.tut.ac.jp
Kenichi Kanatani
Department of Computer Science, Okayama University,
Okayama 700-8530 Japan
kanatani@suri.it.okayama-u.ac.jp

Abstract

A new method is presented for computing the fundamental matrix from point correspondences: its singular value decomposition (SVD) is optimized by the Levenberg-Marquard (LM) method. The search is initialized by optimal correction of unconstrained ML. There is no need for tentative 3-D reconstruction. The accuracy achieves the theoretical bound (the KCR lower bound).

1 Introduction

Computing the fundamental matrix from point correspondences is the first step of many vision applications including camera calibration, image rectification, structure from motion, and new view generation [7].

A popular approach is to do maximum likelihood (ML) computation without imposing the constraint that the fundamental matrix has rank 2; the rank constraint was imposed a posteriori in a statistically optimal manner. The resulting solution has accuracy close to the theoretical bound (KCR lower bound [2, 8]).

Another approach is to do optimization subject to the rank constraint [1, 4, 13, 14]. In this paper, we propose a new method in this line. Following Bartoli and Sturm [1], we optimize the singular value decomposition (SVD) of the fundamental matrix, but there is no need to include 3-D coordinates or camera matrices as unknowns; we use the Levenberg-Marquard (LM) method in the reduced parameter space.

We numerically compare our LM approach with the optimal correction and show that they have complementary properties. The optimal correction is fairly accurate and robust but gradually deteriorates as noise increases. In contrast, LM achieves very high accuracy when started from a good initial value but is likely to fall into local minima if poorly initialized. From this, we conclude that starting LM from the optimal correction achieves the highest accuracy, which agrees with the theoretical bound (the KCR lower bound). The result is also compared with the bundle adjustment approach.

We summarize the mathematical background in Sect. 2. We describe the optimal correction approach in Sect. 3 and our LM approach in Sect. 4. In Sect. 5, we discuss the bundle adjustment in relation to our methods. Sect. 6 shows numerical experiments. We conclude in Sect. 7.

2 Mathematical Background

Fundamental matrix. Given two images of the same scene, a point (x,y) in the first image and the corresponding point (x',y') in the second satisfy the *epipolar equation* [7]

$$(\begin{pmatrix} x \\ y \\ f_0 \end{pmatrix}, \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix} \begin{pmatrix} x' \\ y' \\ f_0 \end{pmatrix}) = 0, \tag{1}$$

where the image origin $(0,0)$ is at the frame center and f_0 is a scaling constant of the order of the image size (this is for stabilizing numerical computation [6]. In our experiments, we set f_0 = 600 pixels). Throughout this paper, we denote the inner product of vectors **a** and **b** by $(\mathbf{a},\mathbf{b})$. The matrix $\mathbf{F} = (F_{ij})$ in Eq. (1) is of rank 2 and called the *fundamental matrix*. If we define

$$\mathbf{u} = (F_{11},F_{12},F_{13},F_{21},F_{22},F_{23},F_{31},F_{32},F_{33})^\top, \tag{2}$$
$$\xi = (xx',xy',xf_0,yx',yy',yf_0,f_0x',f_0y',f_0^2)^\top, \tag{3}$$

Equation (1) can be rewritten as

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

The magnitude of **u** is indeterminate, so we normalize it to $\|\mathbf{u}\| = 1$. If we write N observed noisy correspondence pairs as 9-D vectors $\{\xi_\alpha\}$ in the form of Eq. (3), our task is to estimate the 9-D vector **u** from $\{\xi_\alpha\}$ using Eq. (4).

Covariance matrices. We write $\xi_\alpha = \bar{\xi}_\alpha + \Delta\xi_\alpha$, where $\bar{\xi}_\alpha$ is the true value and $\Delta\xi_\alpha$ the noise term. The covariance matrix of ξ_α is defined by

$$V[\xi_\alpha] = E[\Delta\xi_\alpha\Delta\xi_\alpha^\top], \tag{5}$$

where $E[\cdot]$ denotes expectation over the noise distribution. If the noise in the x- and y-coordinates is independent and of mean 0 and standard deviation σ, the covariance matrix of ξ_α has the form $V[\xi_\alpha] = \sigma^2 V_0[\xi_\alpha]$ up to $O(\sigma^4)$, where

$$V_0[\xi_\alpha] = \begin{pmatrix} \bar{x}_\alpha^2+\bar{x}_\alpha'^2 & \bar{x}_\alpha'\bar{y}_\alpha' & f_0\bar{x}_\alpha' & \bar{x}_\alpha\bar{y}_\alpha & 0 & 0 & f_0\bar{x}_\alpha & 0 & 0 \\ \bar{x}_\alpha'\bar{y}_\alpha' & \bar{x}_\alpha^2+\bar{y}_\alpha'^2 & f_0\bar{y}_\alpha' & 0 & \bar{x}_\alpha\bar{y}_\alpha & 0 & 0 & f_0\bar{x}_\alpha & 0 \\ f_0\bar{x}_\alpha' & f_0\bar{y}_\alpha' & f_0^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ \bar{x}_\alpha\bar{y}_\alpha & 0 & 0 & \bar{y}_\alpha^2+\bar{x}_\alpha'^2 & \bar{x}_\alpha'\bar{y}_\alpha' & f_0\bar{x}_\alpha' & f_0\bar{y}_\alpha & 0 & 0 \\ 0 & \bar{x}_\alpha\bar{y}_\alpha & 0 & \bar{x}_\alpha'\bar{y}_\alpha' & \bar{y}_\alpha^2+\bar{y}_\alpha'^2 & f_0\bar{y}_\alpha' & 0 & f_0\bar{y}_\alpha & 0 \\ 0 & 0 & 0 & f_0\bar{x}_\alpha' & f_0\bar{y}_\alpha' & f_0^2 & 0 & 0 & 0 \\ f_0\bar{x}_\alpha & 0 & 0 & f_0\bar{y}_\alpha & 0 & 0 & f_0^2 & 0 & 0 \\ 0 & f_0\bar{x}_\alpha & 0 & 0 & f_0\bar{y}_\alpha & 0 & 0 & f_0^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{6}$$

In actual computations, we replace the true positions $(\bar{x}_\alpha, \bar{y}_\alpha)$ and $(\bar{x}'_\alpha, \bar{y}'_\alpha)$ by their data (x_α, y_α) and (x'_α, y'_α), respectively (experiments have confirmed that this does not cause noticeable changes in final results).

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

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

where $\mathbf{P}_{\mathscr{U}}$ is the linear operator projecting $\mathscr{R}^9$ onto the domain $\mathscr{U}$ of $\mathbf{u}$ defined by the constraints $\|\mathbf{u}\| = 1$ and $\det\mathbf{F} = 0$; we evaluate the error of $\hat{\mathbf{u}}$ by projecting it onto the tangent space $T_{\mathbf{u}}(\mathscr{U})$ to $\mathscr{U}$ at $\mathbf{u}$.

Geometry of the constraint. The normal to the hypersurface defined by $\det\mathbf{F} = 0$ is $\nabla_{\mathbf{u}} \det\mathbf{F}$. After scale normalization, it is written as

$$\mathbf{u}^\dagger \equiv N[\begin{pmatrix} u_5u_9 - u_8u_6 \\ u_6u_7 - u_9u_4 \\ u_4u_8 - u_7u_5 \\ u_8u_3 - u_2u_9 \\ u_9u_1 - u_3u_7 \\ u_7u_2 - u_1u_8 \\ u_2u_6 - u_5u_3 \\ u_3u_4 - u_6u_1 \\ u_1u_5 - u_4u_2 \end{pmatrix}], \tag{8}$$

where $N[\cdot]$ denotes normalization into unit norm (this vector represents the "cofactor" of $\mathbf{F}$). It is easily seen that the constraint $\det\mathbf{F} = 0$ is equivalently written as

$$(\mathbf{u}^\dagger, \mathbf{u}) = 0. \tag{9}$$

Since the domain $\mathscr{U}$ is included in the unit sphere $\mathscr{S}^8 \subset \mathscr{R}^9$, the vector $\mathbf{u}$ is everywhere orthogonal to $\mathscr{U}$. Hence, $\{\mathbf{u}, \mathbf{u}^\dagger\}$ is an orthonormal basis of the orthogonal complement of the tangent space $T_{\mathbf{u}}(\mathscr{U})$. It follows that the projection operator $\mathbf{P}_{\mathscr{U}}$ in Eq. (7) has the following matrix representation ($\mathbf{I}$ is the unit matrix):

$$\mathbf{P}_{\mathscr{U}} = \mathbf{I} - \mathbf{u}\mathbf{u}^\top - \mathbf{u}^\dagger\mathbf{u}^{\dagger\top}. \tag{10}$$

KCR lower bound. If the noise in $\{\xi_\alpha\}$ is independent and Gaussian with mean $\mathbf{0}$ and covariance matrix $\sigma^2 V_0[\xi_\alpha]$, the following inequality holds for an arbitrary unbiased estimator $\hat{\mathbf{u}}$ of $\mathbf{u}$ [8]:

$$V[\hat{\mathbf{u}}] \succ \sigma^2 \Big(\sum_{\alpha=1}^{N} \frac{(\mathbf{P}_{\mathscr{U}}\xi_\alpha)(\mathbf{P}_{\mathscr{U}}\xi_\alpha)^\top}{(\mathbf{u}, V_0[\xi_\alpha]\mathbf{u})} \Big)_8^-. \tag{11}$$

Here, $\succ$ means that the left-hand side minus the right is positive semidefinite, and $(\cdot)_r^-$ denotes the pseudoinverse of rank r. Chernov and Lesort [2] called the right-hand side of Eq. (11) the *KCR* (*Kanatani-Cramer-Rao*) *lower bound* and showed that Eq. (11) holds up to $O(\sigma^4)$ even if $\hat{\mathbf{u}}$ is not unbiased; it is sufficient that $\hat{\mathbf{u}} \to \mathbf{u}$ as $\sigma \to 0$.

Maximum likelihood. If the noise in $\{\xi_\alpha\}$ is independent and Gaussian with mean $\mathbf{0}$ and covariance matrix $\sigma^2 V_0[\xi_\alpha]$, *maximum likelihood* (ML) estimation of $\mathbf{u}$ is to minimize the

sum of square *Mahalanobis distances*

$$J = \sum_{\alpha=1}^{N} (\xi_\alpha - \bar{\xi}_\alpha, V_0[\xi_\alpha]_2^- (\xi_\alpha - \bar{\xi}_\alpha)), \tag{12}$$

subject to $(\mathbf{u}, \bar{\xi}_\alpha) = 0$, $\alpha = 1, ..., N$. Eliminating the constraint by using Lagrange multipliers, we obtain [8]

$$J = \sum_{\alpha=1}^{N} \frac{(\mathbf{u}, \xi_\alpha)^2}{(\mathbf{u}, V_0[\xi_\alpha]\mathbf{u})}. \tag{13}$$

The ML estimator $\hat{\mathbf{u}}$ minimizes this subject to $\|\mathbf{u}\| = 1$ and $(\mathbf{u}^\dagger, \mathbf{u}) = 0$.

3 Optimal Correction

A common approach to solve this problem is to minimizes Eq. (13) without considering the rank constraint, compute the SVD of the resulting fundamental matrix, and replace the smallest singular value by 0, producing a "closest" matrix of rank 2 in norm [6]. We call this *SVD correction*.

A more sophisticated method is the *optimal correction* [8, 12]. According to the statistical optimization theory [8], the covariance matrix $V[\tilde{\mathbf{u}}]$ of the rank unconstrained solution $\tilde{\mathbf{u}}$ can be evaluated, so $\tilde{\mathbf{u}}$ is moved in the direction of the mostly likely fluctuation implied by $V[\tilde{\mathbf{u}}]$ until it satisfies the rank constraint. The procedure goes as follows [8]:

1. Compute the following 9×9 matrix $\tilde{\mathbf{M}}$:

$$\tilde{\mathbf{M}} = \sum_{\alpha=1}^{N} \frac{\xi_\alpha \xi_\alpha^\top}{(\tilde{\mathbf{u}}, V_0[\xi_\alpha]\tilde{\mathbf{u}})}. \tag{14}$$

2. Compute the normalized covariance matrix $V_0[\tilde{\mathbf{u}}]$ as follows:

$$V_0[\tilde{\mathbf{u}}] = \tilde{\mathbf{M}}_8^-. \tag{15}$$

3. Update the solution $\tilde{\mathbf{u}}$ as follows ($\tilde{\mathbf{u}}^\dagger$ is defined by Eq. (8) for $\tilde{\mathbf{u}}$):

$$\tilde{\mathbf{u}} \leftarrow N[\tilde{\mathbf{u}} - \frac{1}{3} \frac{(\tilde{\mathbf{u}}, \tilde{\mathbf{u}}^\dagger) V_0[\tilde{\mathbf{u}}]\tilde{\mathbf{u}}^\dagger}{(\tilde{\mathbf{u}}^\dagger, V_0[\tilde{\mathbf{u}}]\tilde{\mathbf{u}}^\dagger)}]. \tag{16}$$

4. If $(\tilde{\mathbf{u}}, \tilde{\mathbf{u}}^\dagger) \approx 0$, return $\tilde{\mathbf{u}}$ and stop. Else, update the normalized covariance matrix $V_0[\tilde{\mathbf{u}}]$ in the form

$$\mathbf{P}_{\tilde{\mathbf{u}}} = \mathbf{I} - \tilde{\mathbf{u}}\tilde{\mathbf{u}}^\top, \qquad V_0[\tilde{\mathbf{u}}] \leftarrow \mathbf{P}_{\tilde{\mathbf{u}}} V_0[\tilde{\mathbf{u}}] \mathbf{P}_{\tilde{\mathbf{u}}}, \tag{17}$$

   and go back to Step 3.

Before doing this, we need to solve unconstrained minimization of Eq. (13), for which several methods exist: the *FNS* (*Fundamental Numerical Scheme*) of Chojnacki et al. [3], the *HEIV* (*Heteroscedastic Errors-in-Variable*) of Leedan and Meer [11], and the *projective Gauss-Newton iterations* of Kanatani and Sugaya [9]. Their convergence properties were studies in [9].

4 Direct Search

We now propose a new method for minimizing Eq. (13) subject to the rank constraint. The fundamental matrix $\mathbf{F}$ has nine elements, on which the normalization $\|\mathbf{F}\| = 1$ and the rank constraint $\det\mathbf{F} = 0$ are imposed. So, it has seven degrees of freedom.

Many types of 7-degree parameterization have been proposed in the past. Typical ones are based on epipoles (e.g., [13, 14]), but the resulting expressions are often complicated, and the geometric meaning of the individual unknowns are not clear. This was overcome by Bartoli and Sturm [1], who regarded the SVD of $\mathbf{F}$ as its parameterization. Their expression is compact, and each parameter has its geometric meaning. However, they included, in addition to $\mathbf{F}$, the 3-D positions of the observed feature points, the relative positions of the two cameras, and their intrinsic parameters as unknowns. Using assumed values, they computed tentative 3-D reconstruction and evaluated the reprojection error. Then, they searched the high-dimensional parameter space for the value that minimizes the reprojection error. Since the tentative 3-D reconstruction from two images is indeterminate, they chose the one for which the first camera matrix is in a particular form ("canonical form").

From the underlying geometry, however, we can see that the necessary and sufficient condition for the corresponding points $\{(x_\alpha, y_\alpha)\}$ and $\{(x'_\alpha, y'_\alpha)\}$ to be a projection of "some" 3-D structure is the epipolar equation of Eq. (1) for "some" $\mathbf{F}$, and Eq. (12) describes the corresponding reprojection error. Hence, bundle adjustment for minimizing the reprojection error by assuming "some" 3-D reconstruction is practically equivalent to minimize Eq. (12) (the meaning of "practically" is discussed shortly), so we need not consider indeterminate 3-D reconstruction.

Here, using the parameterization of Bartoli and Sturm [1], we directly minimize Eq. (13) by the Levenberg-Marquard (LM) method. The fundamental matrix $\mathbf{F}$ has rank 2, so its SVD has the form

$$\mathbf{F} = \mathbf{U}\mathrm{diag}(\sigma_1, \sigma_2, 0)\mathbf{V}^\top, \tag{18}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices, and σ_1 and σ_2 are the singular values. Since the normalization $\|\mathbf{F}\|^2 = 1$ is equivalent to $\sigma_1^2 + \sigma_2^2 = 1$, we adopt the following parameterization[1]:

$$\sigma_1 = \cos\theta, \qquad \sigma_2 = \sin\theta. \tag{19}$$

The orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$ have three degrees of freedom each, so they and θ constitute the seven degrees of freedom. However, the analysis becomes complicated if $\mathbf{U}$ and $\mathbf{V}$ are directly expressed in three parameters each (e.g., the Euler angles or the rotations around each coordinate axis). Following Bartoli and Sturm [1], we adopt the "Lie algebraic method": we represent the "increment" in $\mathbf{U}$ and $\mathbf{V}$ by three parameters each. Let ω_1, ω_2, and ω_3 represent the increment in $\mathbf{U}$, and ω'_1, ω'_2, and ω'_3 in $\mathbf{V}$. The derivatives of Eq. (13) with respect to them are as follows (we omit the details):

$$\nabla_\omega J = \mathbf{F}_U^\top \mathbf{X}\mathbf{u}, \qquad \nabla_{\omega'} J = \mathbf{F}_V^\top \mathbf{X}\mathbf{u}. \tag{20}$$

[1] Bartoli and Sturm [1] took the ratio $\gamma = \sigma_2/\sigma_1$ as a variable. Here, we adopt the angle θ for the symmetry. As is well known, it has the value $\pi/4$ (i.e., $\sigma_1 = \sigma_2$) if the principal point is at the origin $(0,0)$ and if there are no image distortions [7, 8].

Here, the matrices $\mathbf{X}$, $\mathbf{F}_U$, and $\mathbf{F}_V$ are defined by

$$\mathbf{X} = \sum_{\alpha=1}^{N} \frac{\xi_\alpha \xi_\alpha^\top}{(\mathbf{u}, V_0[\xi_\alpha]\mathbf{u})} - \sum_{\alpha=1}^{N} \frac{(\mathbf{u}, \xi_\alpha)^2 V_0[\xi_\alpha]}{(\mathbf{u}, V_0[\xi_\alpha]\mathbf{u})^2}, \tag{21}$$

$$\mathbf{F}_U = \begin{pmatrix} 0 & F_{31} & -F_{21} \\ 0 & F_{32} & -F_{22} \\ 0 & F_{33} & -F_{23} \\ -F_{31} & 0 & F_{11} \\ -F_{32} & 0 & F_{12} \\ -F_{33} & 0 & F_{13} \\ F_{21} & -F_{11} & 0 \\ F_{22} & -F_{12} & 0 \\ F_{23} & -F_{13} & 0 \end{pmatrix}, \qquad \mathbf{F}_V = \begin{pmatrix} 0 & F_{13} & -F_{12} \\ -F_{13} & 0 & F_{11} \\ F_{12} & -F_{11} & 0 \\ 0 & F_{23} & -F_{22} \\ -F_{23} & 0 & F_{21} \\ F_{22} & -F_{21} & 0 \\ 0 & F_{33} & -F_{32} \\ -F_{33} & 0 & F_{31} \\ F_{32} & -F_{31} & 0 \end{pmatrix}. \tag{22}$$

The derivative of Eq. (13) with respect to θ is

$$\frac{\partial J}{\partial \theta} = (\mathbf{u}_\theta, \mathbf{X}\mathbf{u}), \tag{23}$$

where we define

$$\mathbf{u}_\theta = \begin{pmatrix} U_{12}V_{12}\cos\theta - U_{11}V_{11}\sin\theta \\ U_{12}V_{22}\cos\theta - U_{11}V_{21}\sin\theta \\ U_{12}V_{32}\cos\theta - U_{11}V_{31}\sin\theta \\ U_{22}V_{12}\cos\theta - U_{21}V_{11}\sin\theta \\ U_{22}V_{22}\cos\theta - U_{21}V_{21}\sin\theta \\ U_{22}V_{32}\cos\theta - U_{21}V_{31}\sin\theta \\ U_{32}V_{12}\cos\theta - U_{31}V_{11}\sin\theta \\ U_{32}V_{22}\cos\theta - U_{31}V_{21}\sin\theta \\ U_{32}V_{32}\cos\theta - U_{31}V_{31}\sin\theta \end{pmatrix}. \tag{24}$$

Adopting Gauss-Newton approximation, which amounts to ignoring terms involving $(\mathbf{u}, \xi_\alpha)$, we obtain the second derivatives as follows (we omit the details):

$$\nabla_\omega^2 J = \mathbf{F}_U^\top \mathbf{X} \mathbf{F}_U, \qquad \nabla_{\omega'}^2 J = \mathbf{F}_V^\top \mathbf{X} \mathbf{F}_V, \qquad \nabla_{\omega\omega'} J = \mathbf{F}_U^\top \mathbf{X} \mathbf{F}_V,$$

$$\frac{\partial J^2}{\partial \theta^2} = (\mathbf{u}_\theta, \mathbf{X}\mathbf{u}_\theta), \qquad \frac{\partial \nabla_\omega J}{\partial \theta} = \mathbf{F}_U^\top \mathbf{X} \mathbf{u}_\theta, \qquad \frac{\partial \nabla_{\omega'} J}{\partial \theta} = \mathbf{F}_V^\top \mathbf{X} \mathbf{u}_\theta. \tag{25}$$

The LM procedure goes as follows:

1. Initialize $\mathbf{F}$ in such a way that $\det \mathbf{F} = 0$ and $\|\mathbf{F}\| = 1$, and express it as $\mathbf{F} = \mathbf{U}\mathrm{diag}(\cos\theta, \sin\theta, 0)\mathbf{V}^\top$.
2. Compute J in Eq. (13), and let $c = 0.0001$.
3. Compute $\mathbf{F}_U$, $\mathbf{F}_V$, and $\mathbf{u}_\theta$ in Eqs. (22) and (24).
4. Compute $\mathbf{X}$ in Eq. (21), the first derivatives in Eqs. (20) an (23), and the second derivatives in Eqs. (25).
5. Compute the following matrix $\mathbf{H}$:

$$\mathbf{H} = \begin{pmatrix} \nabla_\omega^2 J & \nabla_{\omega\omega'} J & \partial \nabla_\omega J / \partial \theta \\ (\nabla_{\omega\omega'} J)^\top & \nabla_{\omega'}^2 J & \partial \nabla_{\omega'} J / \partial \theta \\ (\partial \nabla_\omega J / \partial \theta)^\top & (\partial \nabla_{\omega'} J / \partial \theta)^\top & \partial J^2 / \partial \theta^2 \end{pmatrix}. \tag{26}$$

6. Solve the 7-D simultaneous linear equations

$$(\mathbf{H}+cD[\mathbf{H}])\begin{pmatrix}\omega\\ \omega'\\ \Delta\theta\end{pmatrix}=-\begin{pmatrix}\nabla_\omega J\\ \nabla_{\omega'}J\\ \partial J/\partial\theta\end{pmatrix}, \tag{27}$$

   for ω, ω', and $\Delta\theta$, where $D[\cdot]$ denotes the diagonal matrix obtained by taking out only the diagonal elements.
7. Update $\mathbf{U}$, $\mathbf{V}$, and θ by

$$\mathbf{U}'=\mathscr{R}(\omega)\mathbf{U},\qquad \mathbf{V}'=\mathscr{R}(\omega')\mathbf{V},\qquad \theta'=\theta+\Delta\theta, \tag{28}$$

   where $\mathscr{R}(\omega)$ denotes rotation around $N[\omega]$ by angle $\|\omega\|$.
8. Update $\mathbf{F}$ as follows:

$$\mathbf{F}'=\mathbf{U}'\mathrm{diag}(\cos\theta',\sin\theta',0)\mathbf{V}'^{\top}. \tag{29}$$

9. Let J' be the value of Eq. (13) for $\mathbf{F}'$.
10. Unless $J'<J$ or $J'\approx J$, let $c\leftarrow 10c$, and go back to Step 6.
11. If $\mathbf{F}'\approx\mathbf{F}$, return $\mathbf{F}'$ and stop. Else, let $\mathbf{F}\leftarrow\mathbf{F}'$, $\mathbf{U}\leftarrow\mathbf{U}'$, $\mathbf{V}\leftarrow\mathbf{V}'$, $\theta\leftarrow\theta'$, and $c\leftarrow c/10$, and go back to Step 3.

5 Bundle Adjustment

In the discussion of Sect. 2, there is a subtle point to be clarified. The transition from Eq. (12) to Eq. (13) is exact; no approximation is involved. Although terms of $O(\sigma^4)$ are omitted and the true values are replaced by their data in Eq. (6), it is numerically confirmed that these do not affect the final results in any noticeable way.

However, although the "analysis" may be exact, the "interpretation" is not strict. Namely, despite the fact that Eq. (13) is the (squared) Mahalanobis distance in the ξ-space, its minimization can be ML only when the noise in the ξ-space is Gaussian, because then and only then is the likelihood proportional to $e^{-J/\text{constant}}$. Strictly speaking, if the noise in the image plane is Gaussian, the transformed noise in the ξ-space is no longer Gaussian, so minimizing Eq. (13) is not strictly ML in the image plane.

In order to test how much difference is incurred, we implemented the bundle adjustment in the image plane (we omit the details) and did comparison experiments.

6 Experiments

Figure 1 shows simulated images of two planar grid surfaces viewed from different angles. The image size is 600×600 pixels with 1200 pixel focal length. We added random Gaussian noise of mean 0 and standard deviation σ to the x- and y-coordinates of each grid point independently and from them computed the fundamental matrix by 1) SVD-corrected LS, 2) SVD-corrected ML, 3) CFNS, 4) optimally corrected ML, and 5) LM.

"LS" means least squares (also called "eight-point algorithm" [6]) minimizing $\sum_{\alpha=1}^{N}(\mathbf{u},\xi_\alpha)^2$, which reduces to simple eigenvalue computation [9]. For brevity, we use the shorthand "ML" for unconstrained minimization of Eq. (13), for which we used the

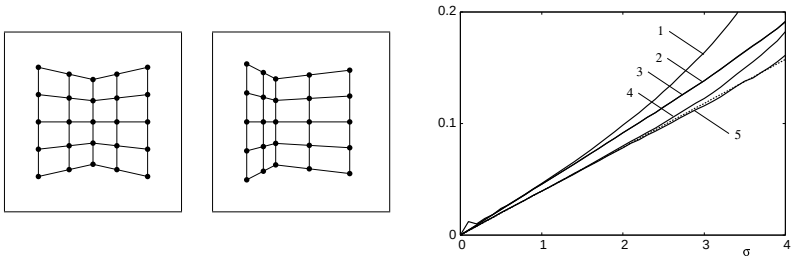


Figure 1: Simulated images of planar grid surfaces and the RMS error vs. noise level. 1) SVD-corrected LS. 2) SVD-corrected ML. 3) CFNS, 4) Optimally corrected ML 5) LM. The dotted line indicates the KCR lower bound.

FNS of Chojnacki et al. [3]. "CFNS" [4] computes the rank-constrained fundamental matrix by repeated eigenvalue computation. The LM and CFNS are initialized by LS. All iterations are stopped when the update of $\mathbf{F}$ is less than 10^{-6} in norm.

On the right Fig. 1 is plotted for σ on the horizontal axis the following root-mean-square (RMS) error D corresponding to Eq. (7) over 10000 independent trials:

$$D = \sqrt{\frac{1}{10000}\sum_{a=1}^{10000} \|\mathbf{P}_{\mathscr{U}}\hat{\mathbf{u}}^{(a)}\|^2}. \tag{30}$$

Here, $\hat{\mathbf{u}}^{(a)}$ is the ath value, and $\mathbf{P}_{\mathscr{U}}$ is the projection matrix in Eq. (10). The dotted line is the bound implied by the KCR lower bound (the trace of the right-hand side of Eq. (11)).

Preliminary observations. We can see that the most popular SVD-corrected LS [6] performs very poorly. We can also see that SVD-corrected ML is inferior to optimally corrected ML [8, 12], whose accuracy is close to the KCR lower bound. The accuracy of LM is nearly the same as optimally corrected ML when the noise is small but gradually outperforms it as the noise increases.

Doing many experiments (not all shown here), we have observed that i) optimally corrected ML is fairly accurate and very robust to noise but gradually deteriorates as noise increases, and ii) LM achieves very high accuracy when started from a good initial value but is likely to fall into local minima if poorly initialized.

The robustness of optimally corrected ML is due to the fact that the computation is done in the redundant ("external") $\mathbf{u}$-space, where J has a simple form of Eq. (13). In fact, we have never experienced local minima in our experiments. The deterioration in the presence of large noise is because linear approximation is involved in Eq. (16).

The fragility of LM is attributed to the complexity of the function J when expressed in seven parameters, resulting in many local minima in the reduced ("internal") parameter space, as pointed out in [13].

Thus, we conclude that the best result is obtained by LM initialized by optimally corrected ML. The CFNS of Chojnacki et al. [4] performs as poorly as SVD-corrected ML. They asserted superiority of CFNS to optimally corrected ML by numerical examples [4], but Fig. 1 contradicts their assertion. The reason for this is fully investigated in [10], where a new method called *EFNS* is proposed to complement CFNS. So, we do not go into the details here.

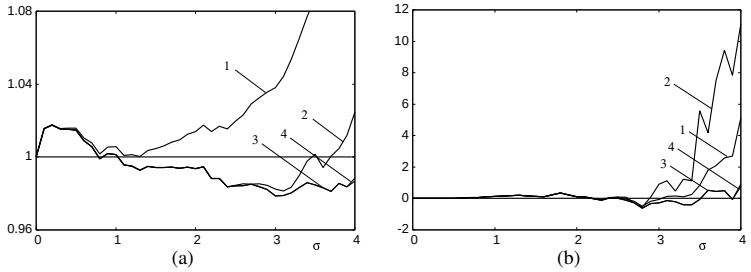


Figure 2: (a) The RMS error relative to the KCR lower bound. (b) The average residual minus $(N-7)\sigma^2$. 1) Optimally corrected ML. 2) LM from LS. 3) LM from optimally corrected ML. 4) Bundle adjustment in the image plane.

	residual	time
1)	45.550	.00052
2)	45.556	.00652
3)	45.556	.01300
4)	45.378	.00764
5)	45.378	.01136
6)	45.378	.01748
7)	45.378	.02580

Figure 3: Left: Real images and 100 corresponding points. Right: Residuals and execution times (sec) for 1) SVD-corrected LS, 2) SVD-corrected ML, 3) CFNS, 4) optimally corrected ML, 5) LM from LS, 6) LM from optimally corrected ML, 7) bundle adjustment in the image plane.

Detailed observations. Figure 2(a) compares 1) optimally corrected ML, 2) LM initialized by LS, 3) LM initialized by optimally corrected ML, and 4) bundle adjustment in the image plane (see Sect. 5). Since the error plots almost coincide if displayed as in Fig. 1, we plot here the ratio D/D_{KCR} of D in Eq. (30) to the corresponding KCR lower bound. Figure 2(b) shows the corresponding residual J (minimum of Eq. (13)). For visual ease, we plot its difference from $(N-7)\sigma^2$, where N is the number of corresponding pairs (J/σ^2 is to a first approximation subject to a χ^2 distribution with $N-7$ degrees of freedom [8], so the expectation of J is approximately $(N-7)\sigma^2$).

We can observe that i) the RMS error of optimally corrected ML increases as noise increases, yet the corresponding residual remains low, meaning that Eq. (13) is very "flat", having nearly the same value around its minimum, ii) LM from LS appears to have high accuracy for noise levels for which the corresponding residual is high, indicating that the solution falls into local minima nearer to the true value, iii) the accuracy of LM improves if initialized by optimally corrected ML, and iv) the result agrees with bundle adjustment in the image plane.

Real image example. Figure 3 shows real images, from which we manually selected 100 pairs of corresponding points and computed the fundamental matrix from them. Since the true value is unknown, we evaluated the final residual J together with the execution time (sec) as listed there. We used Core2Duo E6700 2.66GHz for the CPU with 4GB main memory and Linux for the OS. As expected, SVD-corrected ML is very poor. Again, CFNS performs no better than SVD-corrected ML. Optimally corrected ML and LM (from LS or from optimally corrected ML) arrive at the same solution, which also results from bundle adjustment with a much longer computation time.

7 Conclusions

We presented a new method for computing the fundamental matrix from point correspondences over two images[2]: we adopted the SVD representation of Bartoli and Sturm [1] and optimized it by LM. There is no need for tentative 3-D reconstruction. We concluded that our LM method initialized by optimally corrected ML achieves the accuracy comparable to the KCR lower bound and much time-consuming bundle adjustment.

Acknowledgments: The authors thank Wojciech Chojnacki of the University of Adelaide, Australia for providing

References

[1] A. Bartoli and P. Sturm, Nonlinear estimation of fundamental matrix with minimal parameters, *IEEE Trans. Pattern Anal. Mach. Intell.*, **26**-3 (2004-3), 426–432.

[2] N. Chernov and C. Lesort, Statistical efficiency of curve fitting algorithms, *Comput. Stat. Data Anal.*, **47**-4 (2004-11), 713–728.

[3] W. Chojnacki, M. J. Brooks, A. van den Hengel and D. Gawley, On the fitting of surfaces to data with covariances, *IEEE Trans. Pattern Anal. Mach. Intell.*, **22**-11 (2000-11), 1294–1303.

[4] W. Chojnacki, M. J. Brooks, A. van den Hengel and D. Gawley, A new constrained parameter estimator for computer vision applications *Image Vis. Comput.*, **22**-2 (2004-2), 85–91.

[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] R. I. Hartley, In defense of the eight-point algorithm, *IEEE Trans. Patt. Anal. Mach. Intell.*, **19**-6 (1997-6), 580–593.

[7] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, U.K., 2000.

[8] K. Kanatani, *Statistical Optimization for Geometric Computation: Theory and Practice*, Elsevier Science, Amsterdam, The Netherlands, 1996; Dover, New York, 2005.

[9] K. Kanatani and Y. Sugaya, High accuracy fundamental matrix computation and its performance evaluation, *Proc. 17th British Machine Vision Conf.*, September 2006, Edinburgh, U.K., Vol. 1, pp. 217–226.

[10] K. Kanatani and Y. Sugaya, Extended FNS for constrained parameter estimation, *Proc. 10th Meeting Image Recog. Understand*, Hiroshima, Japan, July 2007.

[11] Y. Leedan and P. Meer, Heteroscedastic regression in computer vision: Problems with bilinear constraint, *Int. J. Comput. Vision.*, **37**-2 (2000-6), 127–150.

[12] J. Matei and P. Meer, Estimation of nonlinear errors-in-variables models for computer vision applications, *IEEE Trans. Patt. Anal. Mach. Intell.*, **28**-10 (2006-10), 1537–1552.

[13] T. Migita and T. Shakunaga, One-dimensional search for reliable epipole estimation, *Proc. IEEE Pacific Rim Symp. Image and Video Technology*, December 2006, Hsinchu, Taiwan, pp. 1215–1224.

[14] Z. Zhang and C. Loop, Estimating the fundamental matrix by transforming image points in projective space, *Comput. Vis. Image Understand.*, **82**-2 (2001), 174–180.

[2]The source code is available at `http://www.iim.ics.tut.ac.jp/~sugaya/public-e.html`

A single-perspective novel panoramic view from radially distorted non-central images

Rana Molana
Computer and Information Science
University of Pennsylvania
molana@grasp.cis.upenn.edu

Kostas Daniilidis
Computer and Information Science
University of Pennsylvania
kostas@cis.upenn.edu

Abstract

In this paper, we propose an image-based technique for panoramic novel-view generation using three uncalibrated wide-angle images as its input. State of the art in novel view generation presumes the calibration and removal of radial distortion or any other deformation resulting from the geometry of a non-central camera. We propose a method which replaces this calibration with the assumption that the epipole corresponding to the novel viewpoint is at the center of radial distortion and that it is known.

1 Introduction

In novel view synthesis, given an image pair, the intensity of each ray in the novel view is determined by finding corresponding point matches between those images, using the epipolar geometries between the two views and with respect to the novel view. This process requires that the given images obey the standard perspective model, with a single viewpoint and no radial distortion. Such assumptions make the creation of novel panoramas a process of acquiring tens of images, each with relatively narrow field-of-view, with special apparatus, and applying calibration and estimation of circular motion [9].

In this paper, we propose an approach where a novel view can be synthesized from multiple views which might be highly radially distorted, and even non-central, without compensating explicitly for the resulting image deformations. We shed new light on the problem (Fig. 1) of "how a scene looks from a scene point?"[8] by eliminating the need for a reference plane and facilitating synthesis of omnidirectional novel views.

We assume that, for $n \geq 3$ views, the location of the epipole of the novel view in each image is known, and, in the case of radially distorted views, that it is coincident with the centre of radial distortion in the image. This assumption is either enforced (for example by choosing a visible scene point as the centre of projection of the view to be synthesized, and actually fixating on this point in the case of radially distorted views), or else it may naturally be satisfied for certain wide-angle imaging systems (for example multiple views of a spherical mirror). Given such a configuration, the 2D star of virtual rays at the novel viewpoint are mapped to a 1D star of lines in each image, which we model as a projection from $\mathbb{P}^2$ (the domain of the novel view) to $\mathbb{P}^1$ (the angles of the line pencil in image plane). We make use of the well known fact that three projections of a line in space (a virtual line in our case) yield a trifocal constraint, from which we extract the projection matrices required for the view synthesis.

The main contributions of this paper are:

- We can use images with arbitrary radial distortion and even non-central cameras, such as imaging the scene via a spherical mirror.
- We show that novel views can be synthesized without recovering the radial calibration or the mapping from pixels to rays in non-central cameras. All steps for the extraction of projection matrices are either linear or in closed form.

Related work: Related work maybe be divided into two categories: (1) approaches concerning calibration and motion estimation with non-central and/or radially distorted cameras; (2) approaches for novel view synthesis from perspective views. The closest approach to our work is the multiple view geometry of 1D radial cameras by Thirthala and Pollefeys [18, 17] where the 1D trifocal tensor is used for radial distortion calibration. Our euclidean calibration from the 1D trifocal tensor is based on the fundamental papers by Quan and Kanade [10, 12] and Faugeras *et al* [2], while Astrom and Kahl [1] further study ambiguity of motion estimation from 1D projections. Our work differs from [17] in the fact that we directly synthesize a novel view without going through the process of removing radial distortion. For the three view case, whereas a purely rotating camera is assumed in [17], we make the assumption of fixation at the viewpoint of the novel view. By avoiding the distortion calibration step, we do not require the given views to be from central cameras, nor that their radial distortion should be rotationally symmetric, and we can produce a novel view without scene reconstructon. Based on this fact, our technique is only tangentially related to recent proposed methods for calibration of non-central cameras under certain scene constraints [13] and estimation of their relative motion.

Methods to calibrate the radial distortion simultaneously with multiple view geometry, where the rays are assumed to be central and without knowledge of the scene structure, include the division model of Fitzgibbon [4], which assumes known centre of radial distortion (as does [17], and as shall we), while the focus of radial expansion method [6] includes the radial distortion centre in the estimation. Grossman *et al* [5] assume a central camera with pure rotational motion but no rotational symmetry in the distortion.

The approach of Irani *et al* [8] has been our motivation for modelling the novel view as a target-to-source mapping rather than an epipolar transfer [3]. Irani's approach, like ours here, differs from classic view synthesis and morphing [14] where the novel view is close to the recorded views. Several approaches exist in the literature regarding creation of panoramic views from new viewpoints by selection of the appropriate rays from a dense lightfield. The dense lightfield can be constructed as a concentric mosaic [9], as well as with arbitrary motion trajectories [19], after estimating the camera trajectory and registering all rays in the same coordinate system.

2 Problem definition

We adopt the following conventions. The angle θ is defined by $\tan\theta = \frac{v}{u}$, where (u,v) are pixel coordinates. For convenience, we use the following shorthand notation for three homogeneous entities, all of which depend only on θ:

$$\mathbf{l}(\theta) \sim \begin{pmatrix} -\sin\theta & \cos\theta & 0 \end{pmatrix}^\top, \mathbf{x}(\theta) \sim \begin{pmatrix} \cos\theta & \sin\theta \end{pmatrix}^\top, \mathbf{n}(\theta) \sim \begin{pmatrix} -\sin\theta & \cos\theta \end{pmatrix}^\top. \quad (1)$$

By definition, $\mathbf{l}(\theta) \in \mathbb{P}^2$ is a line through the image origin, while $\mathbf{x}(\theta), \mathbf{n}(\theta) \in \mathbb{P}^1$ are in the direction of, and normal to, that same radial line. Evidently, $\mathbf{x}(\theta) \sim (u,v)^\top$.

Novel view synthesis from perspective views: We first consider the simplest case when input images are perspective (pinhole) camera views of the scene and we pick any world point as the optical center of the novel single-perspective view. Assuming correspondences $\mathbf{x}_1$ and $\mathbf{x}_2$ in the two input images and epipolar geometries $\mathrm{F}_{NV,1}$ and $\mathrm{F}_{NV,2}$ with respect to the novel view, a point in the novel view can be synthesized as $\mathbf{x}_{NV} = \mathrm{F}_{NV,1}\mathbf{x}_1 \times \mathrm{F}_{NV,2}\mathbf{x}_2$ [3]. Traditional multiple view geometry techniques [7] may be applied to recover the epipolar geometry. Formulating the reverse projection of this epipolar transfer, each novel view ray can be projected into all original views and a color/intensity assigned to the ray by finding the (single) point correspondence that must occur between those image lines. Irani *et al* [8] use this framework to recover a novel view from an arbitrary scene point using the parallax with respect to a plane with known homography.

Novel view synthesis from radially distorted or axial non-central views: The framework above required pre-calibration and removal of radial distortion from the input views; moreover, removal of non-central distortions would only be possible via knowledge (e.g.reconstruction) of the scene [16]. We relax these assumptions and assume that for every input view the epipole to the novel view is the centre of radial distortion. Hence, any radial distortion will be along the projection $\mathbf{l}$ of a virtual ray $\mathbf{d}$. This can apply to non-central projections when rays of the non-central input view intersect a common axis and that axis intersects the novel viewpoint. Although pointwise there is no perspective projection, a virtual ray $\mathbf{d}$ through such a novel viewpoint projects to a straight line $\mathbf{l}$ through the novel view epipole in the image. We can thus model the projection of the 2D star of virtual rays $\mathbf{d}$ as a 1D line pencil in each input view.

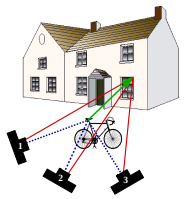


Figure 1. *From three radially distorted, and possibly non-central, views of the scene it is possible to render a novel view of the scene if we assume that the novel view epipole at each of the three input views is the center of radial distortion.*

Although a concurrent novel view epipole and radial distortion centre might seem to require active fixation, some wide angle imaging systems satisfy this automatically. For multiple pinhole camera views of a fixed spherical mirror, captured rays are always non-central [15], yet, regardless of relative camera pose, the sphere centre maps to the radial distortion centre in all views. Thus, choosing the sphere centre as the novel viewpoint fulfils the requirement for epipoles to coincide with the centres of radial distortion for all views. All that remains is to locate the centre of the sphere in each view, achieveable if the perimeter of the mirror is visible in the images. Note that radial distortion will not be rotationally symmetric when the optical axis of the pinhole camera is offset from the centre of the sphere, yet our technique is unaffected since we make no assumptions about the form of the radial distortion.

Assuming a 1D homography between the pencil of planes through each view axis and the pencil of lines in the image, which is indeed the case for most practical "axial" cameras, the mapping of novel view rays ($\mathbb{P}^2$) to image radial line pencil ($\mathbb{P}^1$) is described by a 2×3 homogeneous matrix. (For a perfect pinhole camera this holds at *every* pixel: since it is central, any ray can be picked as the "axis" and a 1D homography is induced on *any* line pencil in the image by the 2D homography of planar perspective projection from pinhole rays to image plane.) We next illustrate by deriving a form for this mapping.

If the rigid body transformation from the novel view coordinate frame to the camera coordinate frame is given by $\mathbf{X}_{\mathrm{C}} = \mathrm{R}\mathbf{X}_{\mathrm{NV}} + \mathbf{T}$, where $\mathrm{R} \in SO(3)$ and $\mathbf{T} \in \mathbb{R}^3$, then the novel view ray with direction $\mathbf{d} \in S^2$ (given in the novel view reference frame) is projected into an input view as the line $\mathbf{l} \in \mathbb{P}^2$ given by

$$\mathbf{l} \sim \mathrm{K}\mathbf{T} \times \mathrm{K}\mathrm{R}\mathbf{d} \quad \text{where K is parameterized as: } \begin{pmatrix} f_u & s & \tilde{u_0} \\ 0 & f_v & \tilde{v_0} \\ 0 & 0 & 1 \end{pmatrix}. \tag{2}$$

The affine transformation K encodes the intrinsic parameters of the mapping from novel view ray ($\mathbf{d}$) to radial line ($\mathbf{l}$). The epipole of the novel view is given by K$\mathbf{T}$; assuming we know its location, we translate our image coordinates to make this epipole our image origin. By doing this, we have set $\mathrm{K}\mathbf{T} \sim (0,0,1)^\top$ in equation (2), and adjusted the variable for the principal point accordingly, giving

$$\mathbf{l} \sim \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \times \mathrm{K}\mathrm{R}\mathbf{d} \quad \Rightarrow \begin{pmatrix} -\sin\theta \\ \cos\theta \\ 0 \end{pmatrix} \sim \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \mathrm{K}\mathrm{R}\mathbf{d} \Rightarrow \mathbf{x}(\theta) \sim \underbrace{\begin{pmatrix} f_u & s & u_0 \\ 0 & f_v & v_0 \end{pmatrix} \mathrm{R}}_{\mathrm{M}} \mathbf{d}. \tag{3}$$

Since M has rank 2, it has a unique RQ-decomposition as the product of a 2×2 homogeneous upper triangular matrix and a 2×3 matrix with orthonormal rows, so the mapping effectively has two intrinsic parameter dofs. Notice that, in a catadioptric case with mirror and pinhole optical axis aligned (i.e., when $\mathbf{T} = (0,0,T_z)^\top$, since the novel view origin also coincides), then $u_0 = v_0 = 0$, and the aspect ratio and skew of the pinhole camera will correspond directly to the two intrinsic parameters of the novel-view-ray-to-radial-line mapping; for that aligned case, any assumptions that can be made about the pinhole camera's aspect ratio and skew parameters can therefore be directly applied to the novel-view-ray-to-radial-line mapping intrinsic parameters.

3 From tri-view correspondences to novel view rays

3.1 Trilinear constraint for radial line correspondences

Given three views, V, V′, V″, we first translate each image, placing the known novel view epipole (assumed to coincide with the radial distortion centre when applicable) at the image origin. Let M, M′, M″ be the 2×3 homogeneous matrices that then map novel view rays to radial lines in respective views. Consider a scene point that is visible in all three views. Let $\mathbf{d} \in S^2$ be the unique novel view ray on which this scene point lies (even if occluded from the novel viewpoint, it still "virtually" coincides with one novel view ray). Let $\mathbf{x}(\theta)$, $\mathbf{x}(\theta')$, $\mathbf{x}(\theta'')$ be the unique radial lines on which the image of this scene point lies in the respective views. Then,

$$\mathbf{x}(\theta) \sim \mathrm{M}\mathbf{d}, \quad \mathbf{x}(\theta') \sim \mathrm{M}'\mathbf{d}, \quad \mathbf{x}(\theta'') \sim \mathrm{M}''\mathbf{d}. \tag{4}$$

Taking the dot product with $\mathbf{n}(\theta), \mathbf{n}(\theta')$ and $\mathbf{n}(\theta'')$ respectively gives

$$\underbrace{\begin{pmatrix} \mathbf{n}(\theta)^\top \mathrm{M} \\ \mathbf{n}(\theta')^\top \mathrm{M}' \\ \mathbf{n}(\theta'')^\top \mathrm{M}'' \end{pmatrix}}_{\mathrm{B}} \mathbf{d} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \tag{5}$$

For this equation to hold the determinant of B must equal zero, giving

$$\det(\mathrm{B}) = \mathbf{n}(\theta)^\top \mathrm{M}\,(\mathrm{M}'^\top \mathbf{n}(\theta') \times \mathrm{M}''^\top \mathbf{n}(\theta'')) = 0, \tag{6}$$

which can be written as

$$\begin{pmatrix} -s\theta s\theta' s\theta'' \\ c\theta s\theta' s\theta'' \\ -s\theta c\theta' c\theta'' \\ c\theta c\theta' c\theta'' \\ s\theta s\theta' c\theta'' \\ -c\theta s\theta' c\theta'' \\ s\theta c\theta' s\theta'' \\ -c\theta c\theta' s\theta'' \end{pmatrix}^\top \underbrace{\begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \end{pmatrix}}_{\mathbf{v}} = 0, \quad \text{with } \mathbf{v} \sim \begin{pmatrix} \mathrm{M}(\mathbf{m}'_1 \times \mathbf{m}''_1) \\ \mathrm{M}(\mathbf{m}'_2 \times \mathbf{m}''_2) \\ \mathrm{M}(\mathbf{m}'_1 \times \mathbf{m}''_2) \\ \mathrm{M}(\mathbf{m}'_2 \times \mathbf{m}''_1) \end{pmatrix}, \text{ where } \mathrm{M}' = \begin{pmatrix} \mathbf{m}'^\top_1 \\ \mathbf{m}'^\top_2 \end{pmatrix} \text{ and } \mathrm{M}'' = \begin{pmatrix} \mathbf{m}''^\top_1 \\ \mathbf{m}''^\top_2 \end{pmatrix}, \tag{7}$$

where $\sin\theta$ and $\cos\theta$ are denoted as $s\theta$ and $c\theta$ respectively. This $\mathbf{v} \in \mathbb{P}^7$ is the 1D trifocal tensor [11, 18]. If $\mathrm{M}, \mathrm{M}', \mathrm{M}''$ and $\mathbf{d}$ satisfy (5) then for any collineation $\mathrm{H} \in PL(3)$ it will be the case that $\mathrm{MH}, \mathrm{M}'\mathrm{H}, \mathrm{M}''\mathrm{H}$ and $\mathrm{H}^{-1}\mathbf{d}$ also satisfy that constraint as:

$$\mathbf{x}(\theta) \sim \mathrm{M}\mathbf{d} = \mathrm{MH}\,\mathrm{H}^{-1}\mathbf{d}, \quad \mathbf{x}(\theta') \sim \mathrm{M}'\mathbf{d} = \mathrm{M}'\mathrm{H}\,\mathrm{H}^{-1}\mathbf{d}, \quad \mathbf{x}(\theta'') \sim \mathrm{M}''\mathbf{d} = \mathrm{M}''\mathrm{H}\,\mathrm{H}^{-1}\mathbf{d}. \tag{8}$$

This 8 dof homography, H, perfectly accounts for the discrepancy between the 7 dofs of $\mathbf{v}$ and the 15 dofs of the three projection matrices it encodes.

3.2 Estimation of the 1D trifocal tensor

For each scene point that lies on a distinct novel view ray, there will be a different corresponding triplet of $\theta, \theta', \theta''$, and each distinct triplet gives a linear homogeneous constraint on $\mathbf{v}$ of the form in equation (7). Given N distinct corresponding triplets of radial lines across three views, the resulting constraints can be stacked into a matrix equation of the form $\mathrm{D}\mathbf{v} = \mathbf{0}$, where D is a N×8 matrix depending only on the measured image angles. If $\mathrm{N} = 7$ then the 1D trifocal tensor, $\mathbf{v}$, can be solved for exactly as the nullspace of D. If $\mathrm{N} > 7$ then $\mathbf{v}$ can be estimated via linear-least squares, for example using SVD. In practice, since this linear estimation requires only 7 correspondence triplets, it is easily and best implemented within a robust framework, such as RANSAC, in order to simultaneously filter the outlier correspondence triplets.

3.3 Determining M, M′ and M″ up to a collineation

The method we use to recover the projection matrices up to right-multiplication by an unknown homography was introduced by Quan and Kanade [12], and is briefly outlined here for completeness. Without loss of generality, we parameterize the three projection matrices as:

$$\mathrm{M} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad \mathrm{M}' = \left(\begin{pmatrix} c_2 \\ -c_1 \end{pmatrix} \underbrace{\begin{pmatrix} \alpha & \beta \end{pmatrix}}_{\mathbf{a}^\top} \quad \underbrace{\begin{pmatrix} c_1 \\ c_2 \end{pmatrix}}_{\mathbf{c}} \right), \quad \mathrm{M}'' = \begin{pmatrix} \mathbf{m}''^\top_1 \\ \mathbf{m}''^\top_2 \end{pmatrix}. \tag{9}$$

Since each matrix is homogeneous, scales of M, M′, M″ are arbitrary; the 10 parameters in $\mathbf{a}, \mathbf{c}, \mathbf{m}''_1, \mathbf{m}''_2$ thus have 7 dofs which are recovereable from the 7 dof 1D trifocal tensor.

Letting μ denote the homogeneous scale factor, the parameterization of the trifocal tensor in terms of the projection matrices, as given in equation (7), can be written as:

$$\underbrace{\begin{pmatrix} 0 & -c_1 & \beta c_2 & 0 & 0 & 0 & -v_1 \\ c_1 & 0 & -\alpha c_2 & 0 & 0 & 0 & -v_2 \\ 0 & 0 & 0 & 0 & -c_2 & -\beta c_2 & -v_3 \\ 0 & 0 & 0 & c_2 & 0 & \alpha c_1 & -v_4 \\ 0 & 0 & 0 & 0 & -c_1 & \beta c_2 & -v_5 \\ 0 & 0 & 0 & c_1 & 0 & -\alpha c_2 & -v_6 \\ 0 & -c_2 & -\beta c_2 & 0 & 0 & 0 & -v_7 \\ c_2 & 0 & \alpha c_1 & 0 & 0 & 0 & -v_8 \end{pmatrix}}_{\mathrm{G}} \begin{pmatrix} \mathbf{m}_1'' \\ \mathbf{m}_2'' \\ \mu \end{pmatrix} = \mathbf{0}. \tag{10}$$

Since $(\mathbf{m}_1''; \mathbf{m}_2''; \mu) \neq \mathbf{0}$, all the 7×7 minors of G must equal zero, yielding two algebraically independent equations, homogeneous in both **a** and **c**, as follows:

$$\underbrace{\begin{pmatrix} c_1v_3 - c_2v_5 & c_1v_4 - c_2v_6 \\ c_1v_7 - c_2v_1 & c_1v_8 - c_2v_2 \end{pmatrix}}_{\mathrm{C}} \underbrace{\begin{pmatrix} c_1^2 + c_2^2 & 0 \\ 0 & c_1 + c_2 \end{pmatrix}}_{\Gamma} \mathbf{a} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{11}$$

These can be solved in closed form for **a** and **c**, up to scale in each case. As neither $c_1 = c_2 = 0$ (which makes M' the null matrix) nor $c_1 = -c_2$ (which makes M' rank 1) is possible, it must be that $\det \mathrm{C} = 0$, which gives

$$\det \mathrm{C} = (v_3v_8 - v_4v_7)c_1^2 + (v_1v_4 + v_6v_7 - v_2v_3 - v_5v_8)c_1c_2 + (v_2v_5 - v_1v_6)c_2^2 = 0. \tag{12}$$

In general, this quadratic has two distinct solutions; since the scale of **c** directly represents the overall scale of M', we arbitrarily fix it to $|\mathbf{c}| = 1$ for both. Then, substituting each **c** solution in equation (11), the respective solutions for **a** are found as the nullspace of $\mathrm{C}\Gamma$. The unknown scale of **a**, corresponds to the arbitrary scale of M, so fix we can fix $|\mathbf{a}| = 1$. In this way, two solutions are found for the matrix M', and for each of these two solutions a unique M'' may be determined from the kernel of G in equation (10).

3.3.1 Discerning the correct solution from the two-fold ambiguity

If we reconstruct the novel view rays for the correspondences, then both solutions will not be within a collineation of the euclidean novel view. Without any further information, it is not possible to distinguish between the two solutions, however in practice many options exist (e.g. a fourth view; information about the scene; an assumption that the radial distortion is constant along conical contours) which would allow for the incompatible solution to be easily identified.

3.4 Estimating the collineation H

When nothing is known regarding the original views' intrinsic parameters or the scene structure, the only option is to pick a collineation that makes the novel view "appear reasonable", for example mapping four of the projective novel rays (reconstructed from correspondences and the projective estimates of M, M', M'') to four desired coordinates.

If information can be gleaned about the scene, or else assumptions made regarding the intrinsic parameters, then sufficient constraints may be derived to determine the particular $\mathrm{H} \in PL(3)$ which gives a euclidean novel view. Details will be case specific; we outline underlying principles for each of these two approaches in the next two subsections.

3.4.1 Using assumptions regarding the original views' intrinsic parameters

Let $\mathrm{M}^{(i)}$, for $i = 1 \ldots n$ views, be the estimated 2×3 projection matrices [1], and let H be the common collineation that upgrades them to the euclidean frame. Consider the following three possible parameterizations for the euclidean $\mathrm{M}^{(i)}\mathrm{H}$:

$$\mathrm{M}^{(i)}\mathrm{H} \sim \bar{\mathrm{K}}^{(i)}\bar{\mathrm{R}}^{(i)} \quad \text{where } \bar{\mathrm{K}}^{(i)} = \begin{pmatrix} \bar{a}_i & \bar{s}_i \\ 0 & 1 \end{pmatrix},\ \bar{\mathrm{R}}^{(i)} \in \mathbb{R}^{2\times3} \text{ with orthonormal rows .} \quad (13)$$

$$\mathrm{M}^{(i)}\mathrm{H} \sim \check{\mathrm{K}}^{(i)}\begin{pmatrix}\check{\mathrm{R}}^{(i)} & \mathbf{t}^{(i)}\end{pmatrix} \quad \text{where } \check{\mathrm{K}}^{(i)} = \begin{pmatrix} \check{a}_i & \check{s}_i \\ 0 & 1 \end{pmatrix},\ \check{\mathrm{R}}^{(i)} \in SO(2),\ \mathbf{t}^{(i)} \in \mathbb{R}^2. \quad (14)$$

$$\mathrm{M}^{(i)}\mathrm{H} \sim \tilde{\mathrm{K}}^{(i)}\mathrm{R}^{(i)} \quad \text{where } \tilde{\mathrm{K}}^{(i)} = \begin{pmatrix} f_{ui} & s_i & u_{0i} \\ 0 & f_{vi} & v_{0i} \end{pmatrix} \text{ and } \mathrm{R}^{(i)} \in SO(3). \quad (15)$$

Unique parameterizations of both forms (13) and (14) will exist for all n euclidean projection matrices [2]. The blatant over-parameterization in (15) is presented as it is more intuitive for the practical case where the final image is from a pinhole camera, perhaps following a mirror reflection, as derived in section 2. Under these parameterizations, the ith camera parameters are related to the common H as follows.

Under parameterization (13) the relation is:

$$\mathrm{M}^{(i)}\mathrm{H}\mathrm{H}^\top\mathrm{M}^{(i)\top} \sim \bar{\mathrm{K}}^{(i)}\underbrace{\bar{\mathrm{R}}^{(i)}\bar{\mathrm{R}}^{(i)\top}}_{I_{2\times2}}\bar{\mathrm{K}}^{(i)\top} = \bar{\mathrm{K}}^{(i)}\bar{\mathrm{K}}^{(i)\top} = \begin{pmatrix} \bar{a}_i^2 + \bar{s}_i^2 & \bar{s}_i \\ \bar{s}_i & 1 \end{pmatrix} \quad (16)$$

This gives two equations per view, relating the elements of $\mathrm{H}\mathrm{H}^\top$ to parameters of that view:

$$\frac{\mathbf{m}_1^{(i)\top}\mathrm{H}\mathrm{H}^\top\mathbf{m}_1^{(i)}}{\mathbf{m}_2^{(i)\top}\mathrm{H}\mathrm{H}^\top\mathbf{m}_2^{(i)}} = \bar{a}_i^2 + \bar{s}_i^2 \qquad \text{and} \qquad \frac{\mathbf{m}_1^{(i)\top}\mathrm{H}\mathrm{H}^\top\mathbf{m}_2^{(i)}}{\mathbf{m}_2^{(i)\top}\mathrm{H}\mathrm{H}^\top\mathbf{m}_2^{(i)}} = \bar{s}_i \quad (17)$$

Under the parameterization (15) the relation is:

$$\mathrm{M}^{(i)}\mathrm{H}\mathrm{H}^\top\mathrm{M}^{(i)\top} \sim \tilde{\mathrm{K}}^{(i)}\underbrace{\mathrm{R}^{(i)}\mathrm{R}^{(i)\top}}_{I_{3\times3}}\tilde{\mathrm{K}}^{(i)\top} = \begin{pmatrix} f_{ui}^2 + s_i^2 + u_{0i}^2 & f_{vi}s_i + u_{0i}v_{0i} \\ f_{vi}s_i + u_{0i}v_{0i} & f_{vi}^2 + v_{0i}^2 \end{pmatrix} \quad (18)$$

(This is the top 2×2 submatrix of the full $\mathrm{K}\mathrm{K}^\top$ - the dual of the image of the absolute conic [7] - for the pinhole camera.) This gives two equations per view, relating the elements of $\mathrm{H}\mathrm{H}^\top$ to the parameters of that view, as follows:

$$\frac{\mathbf{m}_1^{(i)\top}\mathrm{H}\mathrm{H}^\top\mathbf{m}_1^{(i)}}{\mathbf{m}_2^{(i)\top}\mathrm{H}\mathrm{H}^\top\mathbf{m}_2^{(i)}} = \frac{f_{ui}^2 + s_i^2 + u_{0i}^2}{f_{vi}^2 + v_{0i}^2} \quad \text{and} \quad \frac{\mathbf{m}_1^{(i)\top}\mathrm{H}\mathrm{H}^\top\mathbf{m}_2^{(i)}}{\mathbf{m}_2^{(i)\top}\mathrm{H}\mathrm{H}^\top\mathbf{m}_2^{(i)}} = \frac{f_{vi}s_i + u_{0i}v_{0i}}{f_{vi}^2 + v_{0i}^2} \quad (19)$$

Since, H always appears as a symmetric matrix $\mathrm{C} = \mathrm{H}\mathrm{H}^\top$, recovery of H amounts to recovering the 5 parameters of C. Once a positive-definite C is recovered, H can be

[1] Thus far, we assumed $n = 3$ views, but $n > 3$ views could, for example, first be processed as triplets comprising one common view, and then registered using a factorization approach [11].

[2] These two parameterizations follow from the RQ decompostions of a 2×3 and 2×2 matrix, respectively, where in both cases the decomposition is unique because the rank is known to be 2. The assumption that the first 2×2 submatrix has rank 2 across all views is without loss of generality, since the euclidean projection matrices are only up to a similarity, and that similarity can always be chosen to ensure this fact.

found via Cholesky decompositon. In particular, if we know that the parameter ratios are the same across several views, then with just 4 such views we can recover both H and the two common ratios from the RHS of (19). For the case with several randomly posed pinhole camera views of a fixed spherical mirror, (u_{0i}, v_{0i}) will vary for each view, precisely because we translate each image differently to make each epipole the origin. However, since the offset is known for each view, we can write all the (u_{0i}, v_{0i}) in terms of a common unknown principal point and again, provided we have sufficient views, and provided the pinhole camera intrinsics are constant, we can estimate the collineation, albeit non-linearly. If the principal point can be assumed concurrent with the centre of radial distortion (i.e., camera and mirror axes aligned), and pinhole camera skew assumed zero, then 3 such views are sufficient and the solution for C is linear.

3.4.2 Using assumptions regarding the scene

Given projective estimates M, M′, M″, we can projectively reconstruct novel view rays for the correspondence points. Finding the collineation that maps these projective novel rays to a euclidean frame is equivalent to the problem of calibration from a single view, which can be approached variously [7]. If the scene allows identification of two vanishing points (thus horizon of a plane) in the projectively reconstructed novel view, we can upgrade to an unknown affinity. Additionally, if the circular points can be identified, the calibration may be updated up to a similarity.

3.5 Overall algorithm: recovering novel view epipolar geometry

Given three views of a scene such that the novel view epipole in each image is known (and coincides with the centre of radial distortion, where applicable):

1. For each view, translate pixel coodinates so the novel view epipole is the image origin.
2. Compute $N \geq 7$ distinct line correspondence triplets $\mathbf{x}(\theta), \mathbf{x}(\theta')$, $\mathbf{x}(\theta'')$ from point correspondences across the three images.
3. Estimate the best-fit $\mathbf{v}$, via SVD, for the ≥ 7 homogeneous equations of the form in (7), after first filtering outlier correspondences using a robust estimator such as RANSAC.
4. Substitute $\mathbf{v}$ in equation (12) and solve the quadratic to obtain two homogeneous solutions for $\mathbf{c}$. Set $|\mathbf{c}| = 1$ for both solutions.
5. Substitute each of the two solutions for $\mathbf{c}$ in equation (11) and solve to obtain two corresponding homogeneous solutions for $\mathbf{a}$. Set $|\mathbf{a}| = 1$ for both solutions.
6. For each of the two $(\mathbf{a}, \mathbf{c})$-solutions, substitute $\mathbf{v}, \mathbf{a}, \mathbf{c}$ to build the 8×7 matrix G in (10). Extract $(\mathbf{m}''_1, \mathbf{m}''_2)$ from the nullspace of G.
7. Build the two solutions for the projection matrices M, M′, M″ from the two solutions for $(\mathbf{a}, \mathbf{c}, \mathbf{m}''_1, \mathbf{m}''_2)$, according to the parameterization in (9).

Steps for discerning the correct solution from the two-fold ambiguity, and for recovering the collineation to a euclidean novel view, are case specific, as detailed in sections 3.3.1 and 3.4.

4 Rendering the novel view

Let $\mathrm{M}^{(i)}$, for $i = 1 \ldots n$ views, be the 2×3 projection matrices estimated from methods in section 3. If we had enough information to determine the collineation, these projection matrices refer to the euclidean frame and a euclidean novel view will ensue; if not, techniques in this section will lead to a projective novel view.

A sparse novel view from correspondences Let $(\theta_1, \theta_2, \ldots, \theta_m)$ be a correspondence tuplet across $m \geq 2$ of the n views. Then,

$$\begin{pmatrix} \mathbf{n}(\theta_1)^\top \mathrm{M}^{(1)} \\ \vdots \\ \mathbf{n}(\theta_m)^\top \mathrm{M}^{(m)} \end{pmatrix} \mathbf{d} = \mathbf{0}. \tag{20}$$

If $m = 2$ the corresponding novel view ray, $\mathbf{d}$, is the cross-product. For $m > 2$ the nullspace (via SVD) gives $\mathbf{d}$. Using inlier correspondences (including two-view correspondences) from the initial 2D matching for trifocal tensor estimation results in a forward version of novel view synthesis, entailing a short run time but rendering a sparse novel view.

A dense novel view via guided matching Establishing a dense novel view means iterating over all rays to be synthesized, projecting each into all input views. The ray color/intensity can be set from the most photoconsistent pixel along the corresponding radial lines. Ordering should be considered when matching, as multiple points on the same novel ray could be visible in the original views (with a fixed spherical mirror case this not likely: parallax between views due to a non-central caustic [15] would be very small).

5 Preliminary simulations and experiments

Simulations for computing novel view epipolar geometry, using real 3D scene points projected into four synthetic views, are currently being undertaken. A preliminary remark is that correspondences should span > 90 degrees in at least one image for stability. Thus, although the framework itself does not preclude configurations where the novel view epipole/centre of radial distortion is on the perimeter, or even outside the frame, of *all* images (e.g., oblique pinhole views of a spherical mirror slice), such configurations are not ideal. In practical imaging, the radial distortion centre does usually lie within the image, and since wide-angle imaging captures many features, tri-view point correspondences are likely to span the full range of θ in at least one view, making estimation stable. Implementation with a sequence of shots of a spherical mirror, such as in Fig. 2, is future work, but note that correspondences have wide θ span in all images, and that even just three views of a spherical mirror provide more than enough information for a full panoramic novel view, as each visible scene point is captured by at least two images.

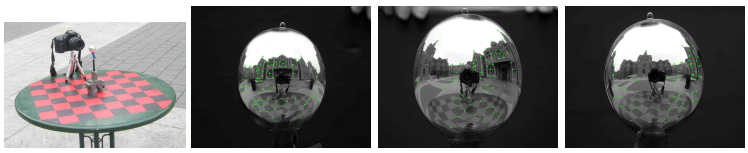


Figure 2. *The recording setup on the left and the three input views.*

6 Conclusion

This paper has proposed a framework for rendering a single-perspective novel panoramic view from radially distorted non-central images, when the epipole of the novel view can be assumed to coincide with the centres of radial distortion in all views. This condition

is *automatically* met by multiple views of a fixed spherical mirror using a pinhole camera (whatever the relative pose), and that is a primary practical use for this framework.

For other wide-angle imaging techniques (using a hand-held camera with a fish-eye lens, say!), precise fixation by the centres of radial distortion on a single scene point in space will be prone to some error, and analysis of sensitivity of the epipolar geometry estimation to errors in the fixation and/or errors in the assumed centres of radial distortion is currently being undertaken, in order to quantify framework limitations. It should be noted that state-of-the-art techniques for generating novel panoramic views typically assume circular motion, or pure rotation, and so our requirement that the views should fixate on a novel viewpoint is actually *less* restrictive.

Rendering a dense novel view will depend on the success of guided matching along the corresponding radial lines. Stereo rectification or, more appropriately, rectification via a homography to one of the original views as in [8], is not a possibility because the intra-view epipolar geometry is not recovered (since no model is assumed for the radial distortion). Stereo rectification to the novel view will not be useful for typical configurations, where the novel view epipole lies within the images. Exploration of techniques for this 1D guided matching is therefore of interest for future work.

References

[1] Kalle Astrom and Fredrik Kahl. Ambiguous configurations for the 1d structure and motion problem. *Journal of Mathematical Imaging and Vision*, 18(2):191–203, 2003.

[2] O. Faugeras, L. Quan, and P. Sturm. Self-calibration of a 1d projective camera and its application to the self-calibration of a 2d projective camera. *PAMI*, 22(10):1179–1185, 2000.

[3] O. Faugeras and L. Robert. What can two images tell us about a third one? In *ECCV*, pages 485–492, 1994.

[4] A. W. Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *CVPR*, pages 125–132, 2001.

[5] E. Grossmann, E. Lee, P. Hislop, D. Nister, and H. Stewenius. Are two rotational flows sufficient to calibrate a smooth non-parametric sensor? In *CVPR*, pages 1222–1229, 2006.

[6] R. Hartley and S.B. Kang. Parameter-free radial distortion correction with centre of distortion estimation. In *ICCV*, pages 1834–1841, 2005.

[7] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[8] M. Irani, T. Hassner, and P. Anandan. What does the scene look like from a scene point? In *ECCV*, pages 883–897, 2002.

[9] G. Jiang, Y. Wei, L. Quan, H. Tsui, and H.Y. Shum. Outward-looking circular motion analysis of large image sequences. *PAMI*, 27(2):271–277, 2005.

[10] L. Quan. Self-calibration of an affine camera from multiple views. *IJCV*, 19(1):93–105, 1996.

[11] Long Quan and Takeo Kanade. A factorization method for affine structure from line correspondences. In *CVPR*, page 803, 1996.

[12] Long Quan and Takeo Kanade. Affine structure from line correspondences with uncalibrated affine cameras. *PAMI*, 19(8):834–845, 1997.

[13] Srikumar Ramalingam, Peter F. Sturm, and Suresh K. Lodha. Theory and calibration for axial cameras. In *ACCV*, pages 704–713, 2006.

[14] S. M. Seitz and C. R. Dyer. View morphing. In *SIGGRAPH*, pages 21–30, 1996.

[15] R. Swaminathan, M. D. Grossberg, and S. K. Nayar. Non-Single Viewpoint Catadioptric Cameras: Geometry and Analysis. *IJCV*, 66(3):211–229, Mar 2006.

[16] R. Swaminathan, M.D. Grossberg, and S.K. Nayar. A Perspective on Distortions. In *CVPR*, volume II, pages 594–601, Jun 2003.

[17] S. Thirthala and M. Pollefeys. Multi-view geometry of 1d radial cameras and its application to omnidirectional camera calibration. In *ICCV*, pages 1539–1546, 2005.

[18] S. Thirthala and M. Pollefeys. The radial trifocal tensor: A tool for calibrating the radial distortion of wide-angle cameras. In *CVPR*, pages 321–328, 2005.

[19] A. Zomet, D. Feldman, S. Peleg, and D. Weinshall. Mosaicing new views: the crossed-slits projection. *PAMI*, 25(6):741–754, June 2003.

Robust Active Appearance Models with Iteratively Rescaled Kernels

M.G. Roberts, T.F. Cootes, J.E. Adams
Department of Imaging Science and Biomedical Engineering
University of Manchester
Manchester, M13 9PL, UK
martin.roberts@manchester.ac.uk

Abstract

Active appearance models (AAMs) are widely used to fit statistical models of shape and appearance to images, and have applications in segmentation, tracking, and classification of structures. A limitation of AAMs is that they are not robust to a large set of gross outliers. Using a robust kernel can help, but there are potential problems in determining the correct kernel scaling parameters. We describe a method of learning two sets of scaling parameters during AAM training: a coarse and a fine scale set. Our algorithm initially applies the coarse scale and then uses a form of deterministic annealing to reduce to the fine outlier rejection scaling as the AAM converges. The algorithm was assessed on two large datasets consisting of a set of faces, and a medical dataset of images of the spine. A significant improvement in accuracy and robustness was observed in cases which were difficult for a standard AAM.

1 Introduction

Active Appearance Models (AAMs) [2] are statistical models of object shape and appearance which have been used to represent and locate objects in a wide range of domains. In their basic form they match a model to a new image using an *analysis-by-synthesis* approach, in which a synthetic image is generated from the model, and then compared to the target image. The model parameters are optimised to synthesise an image as close as possible to the target. One of the strengths of AAMs is that by using a sum-of-squares measure to compare model and target, they can exploit the linear nature of the problem to perform fast parameter updates and thus are able to match to a new image very quickly.

The constraints of the underlying appearance model make AAMs reasonably robust against noise. However, sum-of-squares terms can be badly affected by large outliers. Image regions where the intensities are far from the predicted model values can have a significant deleterious effect, and may lead to poor overall matches or divergence. For example, a model of faces trained on people without beards or glasses may fail badly on images of people with beards or glasses; or on an image where part of the face is occluded. In the medical context there may be a surgical implant in part of the image. A natural method of mitigating such problems is to introduce methods from robust statistics [7]. Typically these seek to downweight larger residuals. However some scaling parameter

is then required to define which residuals are 'large'. One approach to choosing such a parameter is to use estimates of the expected variance of the residuals when matching the model to the training set. However, these tend to be underestimates, and are certainly not appropriate during early stages of search when the model may start far from the true position, leading to large initial residuals.

In this paper we introduce a novel approach in which we use a large initial estimate of the scaling, and gradually reduce the scale as the search algorithm proceeds. So as the image match becomes better, the algorithm becomes stricter as to what it considers inliers. We describe the technique, and give the results of experiments on two different datasets (facial photographs and a medical dataset), demonstrating that the approach leads to improved performance when matching to 'difficult' examples.

2 Background

2.1 Active Appearance Models

Statistical appearance models [2] are linear models of object shape and texture variation, which are learned from a training set. An appearance model is constructed by first learning the statistics of shape and texture variation, and then incorporating both into a set of combined *appearance* parameters **c**. These represent the shape, **x**, and texture, **g**, (in the model frame) according to

$$\begin{aligned} \mathbf{x} &= \bar{\mathbf{x}} + \mathbf{Q}_s \mathbf{c} \\ \mathbf{g} &= \bar{\mathbf{g}} + \mathbf{Q}_g \mathbf{c} \end{aligned} \tag{1}$$

where $\bar{\mathbf{x}}$ is the mean shape, $\bar{\mathbf{g}}$ the mean texture in a mean shaped patch and $\mathbf{Q}_s$,$\mathbf{Q}_g$ are matrices describing the modes of variation derived from the training set.

A shape in the image frame, **X**, can be generated by applying a suitable global transformation (such as a similarity transformation) to the points in the model frame. The texture in the image frame is generated by applying a scaling and offset to the intensities generated in the model frame. These shape pose and texture scaling parameters are concatenated with the appearance model parameters **c** to give the full set of AAM parameters, denoted by **p**.

The AAM seeks to minimise a sum-of-squares problem of the form

$$F(\mathbf{p}) = |\mathbf{r}(\mathbf{p})|^2 = \mathbf{r}^T \mathbf{r} \tag{2}$$

where the vector of residuals **r** is calculated as:

$$\mathbf{r} = \mathbf{w}(I : \mathbf{p}) - \mathbf{g} \tag{3}$$

where $\mathbf{w}(I : \mathbf{p})$ is the (normalised) texture sampled from the image I given the shape defined by model parameters **p**. The AAM is trained (see [2]) how to solve this minimisation by learning the relationship between perturbations in parameters and the texture residuals that these induce. This relationship is then inverted to provide an update matrix that can be applied to the current residual vector. When applied iteratively this provides an efficient model fitting scheme where the necessary parameter change $\delta\mathbf{p}$ to the parameter vector **p**, given **r** is estimated as:

$$\delta\mathbf{p} = -\mathbf{R}\mathbf{r} \tag{4}$$

where $\mathbf{R}$ is derived from the Jacobian $\mathbf{J} = \frac{\delta \mathbf{r}}{\delta \mathbf{p}}$ thus.

$$\mathbf{R} = \left[\mathbf{J}^T \mathbf{J}\right]^{-1} \mathbf{J}^T \tag{5}$$

2.2 Robust Extensions of AAMs

Edwards et al [4] derived an exclusion threshold for each pixel based on the distribution in the training set during AAM matching. Residuals beyond their threshold are ignored in the AAM update step. Stegmann et al [17] extended this by introducing a robust measure for the comparison on AAM solutions, which changes the convergence criterion. However the AAM update steps are still performed as in [4] during the initial search. There are two potential problems with this. Firstly rejecting certain pixels early on can ignore useful information, and may even lead to convergence failure - for example if there is a large initial pose error. Secondly the AAM update matrix with a robust kernel objective function should ideally be changed. In [17] this problem is partly avoided by the use of a simulated annealing phase after conventional AAM search. Gross et al [15] propose a modification that solves this update matrix problem by using a variable Hessian that can be efficiently computed assuming a degree of spatial coherence. Theobald et al [1] experimented with using this method with a variety of kernels. However less attention was given to the problem of determining kernel scaling parameters, which is particularly important in problems where the AAM may not have a good initialisation.

A different approach was adopted by Beichel et al [5]. Their robust AAM identifies outliers by analysing the multi-modal nature of the residual histogram. This method has the advantage of avoiding the direct use of cutoff thresholds - which might be set too low for a particular target image - but at the price of some complexity in the outlier identification method.

Our approach is to recast the objective function as a weighted sum of squares in a way that allows us to apply the standard AAM update to weighted residuals. The scale parameters used for the weighting start off at a coarse scale but are gradually reduced as the AAM converges.

3 Robust Kernel Application and Scaling

We adopt an M-estimator approach adapted to the AAM framework. Given a robust kernel form ρ, the aim is to minimise the objective function:

$$S = \sum_{k=1}^{n} \rho(r_k, \sigma_k) \tag{6}$$

We have used the Geman-McClure [16] kernel for a residual error r_k at pixel k.

$$\rho(r_k, \sigma_k) = \frac{{r_k}^2}{{r_k}^2 + \sigma_k^2}, \tag{7}$$

where σ_k is a scale parameter that controls the convexity of the robust function.

This kernel has some history of use in computer vision, including learning and fitting eigenspace models [9, 13], and in robustly estimating optical flow [14]. The

Geman-McClure robust function is twice-differentiable, and although its influence function (derivative) tends to zero for large errors, there is no absolute cut-off point. The kernel response is always less than the L_2-norm, and the influence is approximately linear over the inlier region $[-\frac{\sigma}{\sqrt{3}}, \frac{\sigma}{\sqrt{3}}]$, after which it declines.

The scaling of the GM kernel is typically some multiple β of the (inlier) standard deviation. The influence of inliers should typically extend to at least one standard deviation, whereas the GM function reduces a residual's influence after $\frac{\sigma}{\sqrt{3}}$. Also because training set inadequacies typically underestimate the true residuals encountered in unseen images we extend the inlier region by a further 50%, and use: $\beta = 1.5\sqrt{3}$.

We note that the conventional AAM is trained to minimise the residual sum of squares - the residuals are not normalised by their variance. Although this is not a maximum likelihood estimator, in practise it appears to work well because the image patch typically includes large numbers of uninteresting pixels with low variance but little structure. So we modify the objective function to make it more like a conventional AAM and optimise:

$$S' = \sum_{k=1}^{n} \sigma_k^2 \rho(r_k, \sigma_k) \tag{8}$$

This form of sum of robust error functions $\rho_k(r_k)$ can be recast as an Iteratively Weighted Least Squares problem:

$$\tilde{S} = \sum_{k=1}^{n} \sigma_k^2 w_k {r_k}^2 \qquad w_k = \frac{1}{r_k}\frac{d\rho_k}{dr_k} \tag{9}$$

with weights determined by the kernel influence, since at the true minimum these problems have the same solution, as can readily be seen by differentiation (and regarding the weights as fixed for the current iteration). We use the AAM update matrix as a quick though approximate solver, applying it to the "kernel-ised" residuals $\mathbf{r}'$, with:

$${r_k}' = \sqrt{w_k}\gamma_k r_k \tag{10}$$

where γ_k is a normalisation constant chosen so that ${r_k}' = r_k$ for a residual at a suitable small proportion ε [1]of the residual standard deviation.

This leads to the following mapping to apply to the residual vector, which we denote by the vector function $\mathbf{K}(\mathbf{r}, \sigma)$

$${r_k}' = [\mathbf{K}(\mathbf{r}, \sigma)]_k = \gamma_k \frac{r_k}{{r_k}^2 + {\sigma_k}^2} \qquad \gamma_k = \frac{\varepsilon^2 + \beta^2}{\beta^2}{\sigma_k}^2 \tag{11}$$

Essentially our robust AAM treats an outlier residual as though it were a smaller residual, which leads to a reduced update step to the model parameters (due to this pixel). However because the implicit Jacobian being used is still $\frac{\delta \mathbf{r}}{\delta \mathbf{p}}$ and not $\frac{\delta \mathbf{r}'}{\delta \mathbf{p}}$, the estimate of the required update step becomes biassed for large outliers. The situation is complicated by having varying biases due to many residuals spread over many parameters. However if we consider the limiting case of the contribution of a specific residual r_k to a single parameter model, we would see by comparing the two Jacobians $\frac{\delta \mathbf{r}}{\delta \mathbf{p}}$ and $\frac{\delta \mathbf{r}'}{\delta \mathbf{p}}$ that

$$\hat{\delta p_k} = \sqrt{w_k}\,\delta {p_k}^{(0)} \tag{12}$$

[1]We used $\varepsilon = 0.1$.

where $\delta\hat{p}_k$ is the update step contribution we use, whereas $\delta p_k{}^{(0)}$ is the one really required in minimising $\tilde{S}$ in 9. Thus we tend to underestimate the step for outliers (where the weights are small) and hence bias the step direction in favour of the inliers (which have weights near unity). The bias is mitigated by the fact that $\sqrt{w_k}$ tends to zero more slowly than w_k itself. This bias would be worse for rapidly decaying kernels, but the Geman-McClure kernel we use is relatively benign to outliers, and a degree of bias towards inliers may be acceptable. We also reduce the effect of bias by gradually reducing the scale as discussed below.

So although the AAM update matrix should ideally be adapted to the weights, the standard AAM update matrix still gives an approximate solution in the inlier region where the weights are close to unity. Nevertheless the underestimate of the update step can cause convergence failures if starting from too far away from the solution, because reducing a large residual can cause the weighting to increase in a way that makes the overall objective function increase (the weight increases faster than the residual squared decreases). In fact such local minima problems can occur in any case with other forms of Iterated Weighted Least Squares. There is always the problem of how to select correct scalings for the $\{\sigma_k\}$ so that outliers are rejected without causing convergence failures on genuine data when the AAM is initialised some way off the target shape.

Our proposed solution is to apply a form of deterministic annealing similar to that used in [9, 13] (which used quasi-Newton non-linear optimisation). We calculate two sets of scaling parameters: $\{\tilde{\sigma}_{Ik}\}$ and $\{\sigma_{Fk}\}$ for the initial and final scale parameters respectively. In the initial AAM updates the scaling is such that the vast majority of residuals are inliers, and so the AAM update equation can validly be used to find an approximate solution to the weighted least squares problem. Then the scale is gradually reduced. The initial scale parameters are learned during the standard (kernel-free) AAM training. We learn the Jacobian of the induced residual vector $\mathbf{r}$ as parameters $\mathbf{p}$ are displaced from their correct values. This Jacobian is used to derive the initial set $\{\tilde{\sigma}_{Ik}\}$ from the modelled parameter variances. However if the AAM has really converged successfully then the final residuals should be substantially lower, and the final residual variances should be approximated by those of the underlying appearance model. These can be learnt by refitting the appearance model back to its own training set [2]. We denote the underlying appearance model ideal standard deviations as $\sigma^{(\mathbf{0})}$. Our approach reduces the kernel scaling parameters from one set to the other. AAMs are typically used with coarse-to-fine search using Gaussian image pyramids, and this reduction is applied at each stage of the image pyramid before moving onto the next.

In [6], the median absolute deviation (MAD) over the entire image dataset was used as a lower bound on each pixel's scale parameter. On the assumption that the training set represents outlier-free data with zero-mean uncorrelated Gaussian distributed residuals, the MAD over all pixels has an expected value of: $\hat{MAD}(\sigma) = \frac{\sqrt{\Sigma_{k=1}^{n}\sigma_k^2}}{1.4826\sqrt{n}}$ where n is the number of pixels.

We obtain estimates of the initial residual standard deviations by summing over the m_p AAM parameters and using the Jacobian $\mathbf{J}$ and the AAM parameter variances $\{\sigma_{j_p}^2\}$. For the appearance model parameters $\mathbf{c}$ these variances are known, and determined via the eigenvalues of the appearance model modes. For the pose parameters (e.g. similarity transform from model to image reference frame) we assume the associated parameter

[2] or preferably a jackknifed miss-1-out estimate can be derived

standard deviation is the positive displacement range used in AAM training (i.e. when deriving the estimate of **J**). Then

$$\sigma_{Ik} = \sqrt{\sum_{j_p=1}^{m_p} {J_{kj_p}}^2 \sigma_{j_p}^2} \tag{13}$$

Applying the MAD lower bound and rescaling by β gives:

$$\tilde{\sigma}_{Ik} = \beta max(\sigma_{Ik}, M\hat{A}D(\sigma_{\mathbf{I}})) \tag{14}$$

Whereas for the converged kernel parameters:

$$\sigma_{Fk} = \beta max(\sigma_k^{(0)}, M\hat{A}D(\sigma^{(\mathbf{0})})) \tag{15}$$

We now summarise the application of the kernel and its rescaling more formally.

For each image pyramid level loop:

1 Set $\sigma = \tilde{\sigma}_I$
2 While the AAM update step improves the solution loop:
  2.1 Calculate the raw residual error vector $\mathbf{r}(\mathbf{p})$
  2.2 Calculate $\mathbf{r}'(\mathbf{p}) = \mathbf{K}(\mathbf{r}, \sigma)$
  2.3 Compute current kernel-ised error norm $E = \|\mathbf{r}'\|^2$
  2.4 Set $\Delta = 1$
  2.5 Update the parameter vector $\tilde{\mathbf{p}} = \mathbf{p} - \Delta \mathbf{R}\mathbf{r}'$
  2.6 Calculate a new error vector $\tilde{\mathbf{r}}$ using the updated parameters $\tilde{\mathbf{p}}$
  2.7 Compute the updated kernel-ised error vector $\tilde{\mathbf{r}}' = \mathbf{K}(\tilde{\mathbf{r}}, \sigma)$
  2.8 If $\|\tilde{\mathbf{r}}'\|^2 < E$, then accept the new parameters: $\mathbf{p} = \tilde{\mathbf{p}}$
  2.9 Else retry for $\Delta = 1.5, \Delta = 0.5, \Delta = 0.25$ etc and go to step 2.5; or exit this loop when no more improvement results
  2.10 End loop
3 Calculate the median reduction ratio μ of $\{\frac{\sigma_{Fk}}{\sigma_{Ik}}\}$
4 Given annealing decay constant α (e.g.$\alpha = 0.9$), set the number of kernel rescaling steps $N = \left\lceil \frac{log(\mu)}{log(\alpha)} \right\rceil$
5 for $N+1$ rescalings do
  5.1 if final rescaling set $\sigma = \sigma_{\mathbf{F}}$
  5.2 else: for each k: $\sigma_k = max(\sigma_{Fk}, \alpha\sigma_k)$
  5.3 Set $M = 4$ (or other suitable update count)
  5.4 Set $\Delta = \frac{1}{M}$
  5.5 for M forced updates do
    5.5.1 Recalculate the raw residual error vector $\mathbf{r}(\mathbf{p})$
    5.5.2 Calculate $\mathbf{r}'(\mathbf{p}) = \mathbf{K}(\mathbf{r}, \sigma)$
    5.5.3 Update the parameter vector $\tilde{\mathbf{p}} = \mathbf{p} - \Delta \mathbf{R}\mathbf{r}'$
    5.5.4 $\mathbf{p} = \tilde{\mathbf{p}}$
  5.6 end loop

6 end loop

7 Repeat AAM update loop (2) at reduced scale now $\sigma = \tilde{\sigma}_F$

8 Move to next image level

End loop over image pyramid hierarchy

4 Experiments

We used a standard set of face images [10], of size 720x576 pixels. An AAM was trained using a subset of the data containing 162 subjects free of facial hair (beards) and glasses. The form of appearance model used was a feature AAM using sigmoidally renormalised corner and edge features combined with similarly renormalised cartesian gradients. See [8] for details. This has been shown to generally outperform a standard texture AAM. The renormalisation to the local image feature statistics also means that there are less likely to be problems with the kernel scaling being inappropriate to a particular image.

The test set used was the remainder of 136 subjects having either facial hair or wearing glasses. Thus the appearance model would be expected to be unable to accurately fit to significant portions of the test cases. This will typically cause a number of overall convergence failures. We tested running the AAM with and without the robust kernel, to see if the robust kernel improves convergence reliability in cases which significantly depart from the trained model. There were 3 images used for each subject, giving a total test set size of 408. The AAM was fitted to each test case using a fixed grid of initial position and scale offsets. The initial pose parameters were set by performing an initial best fit of the shape model to the (known) test case points. Then the appearance parameters were reset to zero, and the x,y and scale pose parameters varied as follows. The x and y values were stepped respectively between [-20,20] pixels with a step-size of 5 pixels; the scale was stepped between 1/1.05,1.0,1.05 respectively. This gives a total intialisation grid of 9x9x3=243 searches for each test image.

For the medical images miss-8-out experiments were performed on a dataset of 308 DXA images of the spine. These used 10 initialisations for each image, based on random perturbations on the mean shapes implied by the vertebral centres. See [12] for details. DXA images use a very low X-ray dose, which makes them quite noisy. The aim was to locate the positions of the 10 vertebrae from lumbar L4 to thoracic T7. The segmentation algorithm uses multiple AAMs, each of which fits a triplet of vertebrae. These are overlapped, which establishes linkage between the various sub-models. This gives good accuracy on healthy vertebrae [12]. However the main aim of the DXA imaging is in the diagnosis of osteoporosis, as the the weakening of bone leads to fracture of vertebrae as one of the early symptoms of the disease. Accuracy results on severely fractured vertebrae are much poorer, partly because the AAM is prone to miss the fractured vertebrae and instead fit to neighbouring vertebrae. For example the top of a fractured T8 may be fitted to the bottom of the neighbouring T7, which provides a locally similar edge. See Fig 1c for an example. In [11] we used an alternative fractured AAM initialisation which partly resolved this problem. Also the low bone density in fractured vertebrae can give them a very poor signal to noise ratio, which means that noise outliers have a more significant overall effect. We re-ran the experiments of [11] to see if the robust kernel might improve the accuracy on these difficult fractured vertebrae.

	Standard AAM			Robust Kernel AAM		
Error Measure	Mean	Median	90%-ile	Mean	Median	90%-ile
Point-to-Curve	4.93	3.36	10.54	4.64	3.28	8.86
CoG X	1.95	1.05	4.63	1.48	0.94	2.96
CoG Y	3.79	1.22	12.55	3.17	1.08	9.51

Table 1: Search accuracy results (pixels) for the face data. CoG X and CoG Y are the absolute errors in the centre of gravity of the face

	Standard AAM				Robust Kernel AAM			
Fracture Type	Mean (mm)	Median (mm)	90%-ile (mm)	%ge Edges Misfit	Mean (mm)	Median (mm)	90%-ile (mm)	%ge Edges Misfit
Normal	0.73	0.55	1.57	0.20%	0.78	0.59	1.62	0.20%
Grade 1	0.83	0.57	1.77	4.07%	0.83	0.60	1.79	1.61%
Grade 2	1.01	0.67	2.34	4.37%	0.98	0.67	2.26	1.96%
Grade 3	1.65	0.88	4.52	16.64%	1.28	0.78	3.05	8.42%

Table 2: Search accuracy results by vertebral fracture status for spine DXA images. NB Fracture grades 1-3 correspond to height reduction thresholds of 20%, 25% and 40% respectively. Errors are in mm but the %ge edges misfit columns give the percentage ratio of superior or inferior edges fitted erroneously onto a neighbouring vertebra

5 Results

For the face experiments we calculated the mean, median and 90%-ile for: a)mean absolute point-to-curve error over each face image; b) face centre of gravity x and y positions. For the vertebral data we similarly calculated the mean, median and 90%-ile on raw point to curve errors, and also enumerated the proportion of vertebral edges that were erroneously fitted to a neighbouring vertebra. These were analysed according to their classification as normal (non-osteoporotic), or one of 3 fracture severity grades according to a standard system used by radiologists. Results for the face and vertebral datasets are summarised in tables 1 and 2 respectively.

6 Discussion and Conclusions

In the face experiments a statistically significant improvement occurs in mean point-to-curve error, and also in the mean error in the cartesian coordinates of the face centre. Moreover the tail of the distribution appears to be reduced by the use of the robust kernel. The 90th percentile is reduced from 10.54 pixels to 8.86. The median error is also marginally lower, indicating that the improvement in mean error is not at the expense of a general deterioration in the better fitting cases. There is a larger improvement in the accuracy of the estimate of the centre of the face, both in terms of mean and the 90th percentile tail. Overall in the face data there is a modest but significant improvement in accuracy. An example is shown in Figure 1, which shows a partial search failure without the use of the kernel. The AAM appears to have been confused by the facial hair and does not correctly locate the mouth and chin, whereas in the corresponding case with the

Figure 1: Examples. The left (Fig 1a) and middle (Fig 1b) figures show an example from the face data using a standard AAM (leftmost) and with the kernel AAM (middle). The facial hair appears to confuse the non-robust AAM (left) as to the location of the mouth and chin, whereas with the robust kernel (middle) this problem is avoided. The right-hand figure (Fig 1c) shows an edge misfit (standard non-robust AAM) on a vertebra for a grade 3 fracture. The fitted standard 6-point morphometric corner and mid-points are shown. The arrows indicate where the AAM search fitted the top of T8 to the bottom of T7, and the anterior bottom corner of T8 to the top of T9.

kernel applied the solution has converged with reasonable overall accuracy.

In the medical image data, there was no improvement with normal vertebra. In fact there was a mild deterioration when applying the kernel, from 0.73mm to 0.78mm, though this difference is on the margins of statistical significance, and is far smaller than the precision of the underlying manual annotation. There is little difference in mean accuracy on mildly fractured vertebrae, although the number of incorrectly converged solutions on neighbouring vertebra does appear to reduce. But when we come to the challenging case of the severely fractured vertebra there is a substantial reduction in the number of incorrect solutions on neighbouring edges, and this is accompanied by a significant improvement in mean accuracy from 1.65mm to 1.28mm, and the 90%ile error is reduced from 4.52mm to 3.05mm. Where the AAM was already performing well (i.e. the normal vertebrae), there is no advantage to using the kernel on this dataset. However in this application the marginal deterioration in well localised healthy vertebra is an acceptable price for a substantial improvement in accuracy for the more difficult severely osteoporotic cases. The marginal deterioration on normals may be because the use of the standard AAM update matrix on the transformed residuals leads to small biases. In fact variations in $\mathbf{R}$ across the population is a general problem for AAMs. Cootes introduced the Updating AAM [3] to address this. In future we intend combining our form of robust AAM with the Updating AAM, which should lead to better optimisation of the objective function $\tilde{S}$ (equation 9).

In conclusion the use of a Geman-McClure robust kernel prior to applying the standard AAM update matrix leads to a modest but significant improvement in accuracy, when applied to test images which would be expected to provide difficult cases. Initial coarse kernel scaling parameters can be derived using the AAM Jacobian, and we have shown a

method of reducing these to the finer scale of the appearance model itself. This generally avoids the trap of local minima that would be caused by the premature rejection of too many useful pixels. This method could be used with other kernels, but approximations in the update step mean it would work best with kernels having a gradual outlier rejection - e.g. inverse polynomical decay rather than exponential.

References

[1] Theobald BJ, Matthews I, and Baker S. Evaluating error functions for robust active appearance models. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 149–154, 2006.

[2] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:681–685, june 2001.

[3] TF Cootes and CJ Taylor. An algorithm for tuning an active appearance model to new data. In *British Machine Vision Conference*, pages 919–928, 2006.

[4] G.J. Edwards, T. F. Cootes, and C. J. Taylor. Advances in active appearance models. In *7th International Conference on Computer Vision*, pages 137–142, 1999.

[5] Beichel R et al. Robust active appearance models and their application to medical image analysis. *IEEE Transactions on Medical Imaging*, 24:1151–1169, 2005.

[6] Black MJ et al. Robust anisotropic diffusion. *IEEE Transactions on Image Processing*, 7:421–432, 1998.

[7] JP Hampel et al. *Robust Statistics: an Approach based on Influence Functions*. Wiley, 1986.

[8] Scott IM, Cootes TF, and Taylor CJ. Improving active appearance model matching using local image structure. In *18th Conference on Information Processing in Medical Imaging*, pages 258–269, 2003.

[9] De la Torre F and Black MJ. A framework for robust subspace learning. *International Journal of Computer Vision*, 54:117–142, 2003.

[10] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre. Xm2vtsdb: The extended m2vts database. In *Proc. 2nd Conf. on Audio and Video-based Biometric Personal Verification*. Springer Verlag, 1999.

[11] Roberts MG, Cootes TF, and Adams JE. Improving the segmentation accuracy of fractured vertebrae with dynamically sequenced active appearance models. In *MICCAI Joint Disease Workshop*, pages 1–8, 2006.

[12] Roberts MG, Cootes TF, and Adams JE. Vertebral morphometry: semi-automatic determination of detailed shape from dual-energy x-ray absorptiometry images using linked active appearance models. *Investigative Radiology*, 41(12):849–859, 2006.

[13] Black MJ and Jepson AD. Eigentracking: Robust matching and tracking of objects using view-based representation. *International Journal of Computer Vision*, 26(1):63–84, 1998.

[14] Black MJ and Anandan P. A framework for the robust estimation of optical flow. In *IEEE International Conference on Computer Vision*, pages 231–236, 1993.

[15] Gross R, Matthews I, and Baker S. Constructing and fitting active appearance models with occlusion. In *IEEE Workshop on Face Processing in Video*, pages 72–79, 2004.

[16] Geman S and McClure D. Statistical methods for tomographic image reconstruction. *Bulletin of the International Statistical Institute*, LII:4–5, 1997.

[17] M. B. Stegmann, R. Fisker, and B. K. Ersbll. Extending and applying active appearance models for automated, high precision segmentation in different image modalities. In *Scandinavian Conference on Image Analysis*, pages 90–97, 2001.

Unsupervised Learning of Shape Manifolds

Nasir Rajpoot † Muhammad Arif ⋆ Abhir Bhalerao †

† Department of Computer Science, University of Warwick, CV4 7AL, UK
⋆ Pakistan Institute of Engineering and Applied Sciences, Nilore, Pakistan

email: {nasir,arif,abhir}@dcs.warwick.ac.uk

Abstract

Classical shape analysis methods use principal component analysis to reduce the dimensionality of shape spaces. The basic assumption behind these methods is that the subspace corresponding to the major modes of variation for a particular class of shapes is linearised. This may not necessarily be the case in practice. In this paper, we present a novel method for extraction of the intrinsic parameters of multiple shape classes in an unsupervised manner. The proposed method is based on learning the global structure of shape manifolds using diffusion maps. We demonstrate that the method is effective in separating the members of different shape classes after embedding them into a low-dimensional Euclidean space.

1 Introduction

The shape of an object plays a critical role in its representation and can be used as an important cue for indexing purposes for image and video databases, or for searching of a particular phenomenon (for example, regular-shaped cells) in biomedical images, just to name a few applications. Intuitively, an object's shape can be related to the form or structure of the object. A working definition of shape adopted by most researchers in computer vision is *the quality of an object that is invariant to transformations such as translation, rotation, and scaling and to changes in brightness* [13]. An interesting variation in this definition is related to rigid transformations of parts of an object (for instance, fingers of a hand). A desirable representation of shape should *quotient out* all such transformations [9]. However, such a representation is likely to induce the so-called *curse of dimensionality* problem. To circumvent this problem, classical shape analysis methods such as [4, 12, 2] utilise principal component analysis (PCA) to reduce the dimensionality of the problem. One of the basic assumptions that these methods make is that the subspace corresponding to the major modes of variation for a particular class of shapes is linearised. The reality, however, is that such subspaces may be far from linear and hence a non-linear dimensionality reduction method may be required.

It would not be an exaggeration to say that non-linear dimensionality reduction is an active area of research [16]. A range of non-linear dimensionality reduction methods have been proposed in recent years which can be categorised broadly as *local* and *global*. *Local* embedding methods such as Laplacian eigenmaps and Locally Linear Embedding (LLE) attempt to preserve the properties of a small neighbourhood in a high-dimensional space while *global* techniques such as Isomap, Kernel PCA, diffusion maps [3] attempt

to preserve global properties of the data (see [16] for a comprehensive survey on dimensionality reduction methods). Local techniques are computationally efficient and perform well when the local geometry of the manifold is nearly Euclidean while global techniques preserve global structure of the data more faithfully [16].

A few recent studies on the problem of manifold learning with regard to shape analysis and its applications to computer vision can be found in the literature. He *et al.* [7] utilised the manifold learning of images for indexing images and using relevance feedback in the retrieval of images. Souvenir and Pless [14] proposed a manifold learning algorithm for weighted point sets, the idea being that members of different shape classes may be given different weights for efficiently separating intersecting manifolds of shape classes. Yankov and Keogh [17] presented a method for shape clustering using Isomaps for non-linear dimensionality reduction and used the classical Expectation-Maximisation (EM) algorithm for clustering the feature vectors of a reduced dimensionality. More recently, Etyngier *et al.* [6] extended the Laplacian eigenmap algorithm to enable interpolation between shape samples using local weighted means. This results in an elegant algorithm which was applied to both synthetic and real images showing its potential for image segmentation. Twining and Taylor [15] also presented a kernel PCA based extension of the active shape models.

In this paper, we propose a novel method for extracting the intrinsic parameters of multiple shape classes in an unsupervised manner, based on learning the global structure of shape manifolds with the help of diffusion maps. Assuming that similar shapes (and their transformations) belonging to a particular class lie on a sub-manifold in a high-dimensional Euclidean space $\Re^N$, the problem of extracting intrinsic shape parameters reduces to the problem of finding a low-dimensional embedding of the sub-manifold. A concept diagram of our method is shown in Figure 1. We choose the recently proposed

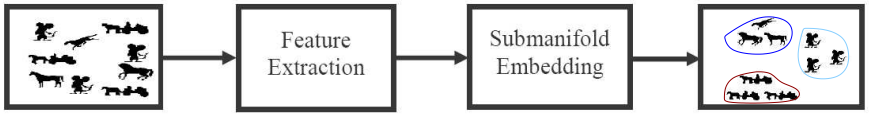


Figure 1: Concept diagram of the proposed method

diffusion maps [3] for global learning of our shape manifolds in an unsupervised manner, due to its control over the dimensionality of the sub-manifolds to embed in the lower-dimensional space. A diffusion map uses diffusion processes to understand the geometry of the high-dimensional manifold (shape manifold, in our case) by defining a Markov random walk on a graph of the data set. Like other spectral clustering methods, it uses eigenfunctions of a similarity-type matrix, termed as the Markov matrix (described in Section 3), as coordinates of the data set in a low-dimensional space. However, it differs from the other aforementioned methods in two respects: (1) it defines a robust metric on the data that reflects the connectivity of data set, and (2) it defines a time parameter in a Markov chain that determines the dimensionality of the sub-manifold [10].

The main contribution of this paper is the presentation of a framework for learning the shape manifolds for multiple classes (demonstrated by using samples from six shape classes; most shape manifold clustering methods listed above are limited to 2 or 3 classes) in an unsupervised manner. The proposed method for extraction of intrinsic shape parameters is shown to be effective using only two dimensions, even in the presence of noise. Moreover, the feature vectors used to represent shapes are invariant to scale and rotation, so no explicit alignment of shapes is required, saving valuable computational time.

The extraction of features from shapes defined as closed curves is outlined in the next section. Embedding of shape sub-manifolds in a low-dimensional space using diffusion maps is described in Section 3. Experimental results are presented and discussed in Section 4, followed by concluding remarks.

2 Shape Representation

As mentioned above, a desirable representation of shape should factor out any translation, scale, and rotation transformations. In order to learn the structure of shape manifolds of different classes in an unsupervised manner, it is crucial that such a representation employs features related to a shape that are associated with main shape characteristics. Several features for shape representation (also known as *shape descriptors*) have been proposed in the literature. It is beyond the scope of this paper to list and compare them all.

Let us regard a shape as a closed curve (*contour*) and initially represent it as a set of boundary points corresponding to the contour. We have chosen *Fourier descriptors* to represent a given shape contour. Fourier descriptors are simple yet powerful means of extracting shape features. It is worth noting that the choice of Fourier descriptors was made with simplicity in mind. The major objective of this study is to investigate the unsupervised learning of shape manifolds; no claim of optimality is made here about the representation of shapes using Fourier descriptors. Having noted that, one of the advantages of using Fourier descriptors is that they can easily be made invariant to any scale and rotation transformations (as described in the next paragraph). This implies that no explicit shape normalisation, as is the case with most well-known shape analysis methods such as [4], is required.

For a given shape, we extract its corresponding contour $\mathscr{C}$ which consists of N (fixed) boundary points (x_i, y_i), for $i = 1, 2, \ldots, N$. The number of boundary points remains fixed by employing cubic spline interpolation for re-sampling of the boundary points. A centroidal distance function r_i, for $i = 1, 2, \ldots, N$, is computed as follows [8, 11],

$$r_i = \sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2} \tag{1}$$

where $(\bar{x}, \bar{y})$ denotes the coordinates of the centroid of the object. The distance vector $\mathbf{r} = \{r_1, r_2, \ldots, r_N\}$ is transformed into frequency domain using FFT. The feature vector $\mathbf{f}$ can then be derived as follows,

$$\mathbf{f} = \left(\frac{|F_1|}{|F_0|}, \frac{|F_2|}{|F_0|}, \ldots, \frac{|F_{N/2}|}{|F_0|} \right)^T \tag{2}$$

where $|F_i|$ denotes the ith Fourier coefficient, with F_0 being the DC component. In the above equation, taking the magnitude of the coefficients yields rotation invariance and their division by $|F_0|$ results in scale invariance. In order to test the robustness of a shape clustering algorithm in the presence of noise, we add Gaussian noise n with zero mean and a standard deviation σ to the boundary points (x_i, y_i) as follows [8],

$$\begin{aligned} x_i' &= x_i + d_i \cdot n \cdot \sin\theta_i \\ y_i' &= y_i + d_i \cdot n \cdot \cos\theta_i \end{aligned} \tag{3}$$

for $i = 1, 2, \ldots, N$, where θ_i is the tangent angle at (x_i, y_i) and d_i denotes the distance between (x_i, y_i) and (x_{i-1}, y_{i-1}). Figure 2 shows an example contour and its corresponding centroidal distance vector without noise and with two different levels of noise.

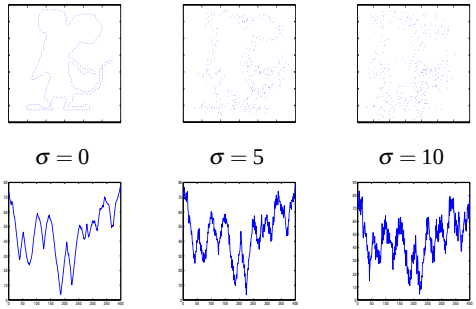


Figure 2: The contour for a Rat image with $N = 400$ boundary points (*Top row*) and the corresponding centroidal distance vector (*Bottom row*) for three levels of noise.

3 Learning the Shape Manifolds with Diffusion Maps

3.1 Computation of Shape Distances

The first step of diffusion maps based framework is to consider the feature vectors $\Omega = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n\}$ corresponding to a data set of various shapes, where n denotes the total number of shapes in the data set. Like most manifold learning methods, we regard the feature vectors as nodes of an adjacency graph (node i representing the feature vector $\mathbf{f}_i$) and compute pairwise similarity matrix between $\mathbf{f}_i$ and $\mathbf{f}_j$, $i, j = 1, 2, \ldots, n$, using Gaussian kernel with width ε,

$$w(\mathbf{f}_i, \mathbf{f}_j) = \exp\left(-\frac{||\mathbf{f}_i - \mathbf{f}_j||^2}{2\varepsilon}\right) \tag{4}$$

A Markov random walk is then defined on the graph by defining the state-transition matrix $\mathbf{P}$ (also known as the Markov matrix), with its (i, j)th element p_{ij} given as follows

$$p_{ij} = \frac{w(\mathbf{f}_i, \mathbf{f}_j)}{d(\mathbf{f}_i)} \tag{5}$$

where $d(\mathbf{f}_i)$ denotes the degree of node $\mathbf{f}_i$ in the graph and is calculated as follows

$$d(\mathbf{f}_i) = \sum_{\mathbf{z} \in \Omega} w(\mathbf{f}_i, \mathbf{z}). \tag{6}$$

It can be seen from the above definitions of the Markov matrix that its ith row $\mathbf{P}_i = \{p_{ij},\ j = 1, 2, \ldots, n\}$ corresponds to the probabilities of transition from node i to all the other nodes after one time step. It is worth noting that unlike most similarity matrices used in other manifold learning methods, this matrix is generally not symmetric.

3.2 Embedding in a Low-Dimensional Space

Once the Markov matrix for a given set of shapes belonging to different classes has been determined, sub-manifolds corresponding to the shape classes can be embedded in a lower-dimensional space as follows. If $\{\lambda_i\}$ is the sequence of eigenvalues of $\mathbf{P}$ such that $|\lambda_0| \geq |\lambda_1| \geq \cdots$, and $\{\psi_i\}$ are the corresponding eigenvectors, then a mapping from

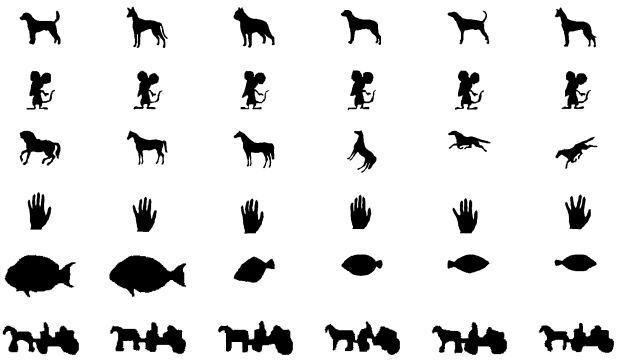


Figure 3: Some of the images used in our experiments.

the shape feature set Ω to a lower-dimensional Euclidean space $\Re^m$, where $m << N$ is the dimensionality of the lower-dimensional subspace, is given by (see [10, 3] for details),

$$\Psi : \mathbf{f} \longmapsto (\lambda_1 \psi_1(\mathbf{f}), \lambda_2 \psi_2(\mathbf{f}), \ldots, \lambda_m \psi_m(\mathbf{f})) \tag{7}$$

The above relationship gives the initial low-dimensional embedding of the shape submanifolds at $t = 1$. In the spirit of a Markov chain, the mapping Ψ evolves as the chain progresses. The above mapping at time $t > 1$ can be computed by multiplying the diffusion coordinates $\psi_i(\mathbf{f})$, $i = 1, \ldots, n$, by the $(\lambda_i)^t$ as given below [3],

$$\Psi^{(t)} : \mathbf{f} \longmapsto \left((\lambda_1)^t \psi_1(\mathbf{f}), (\lambda_2)^t \psi_2(\mathbf{f}), \ldots, (\lambda_m)^t \psi_m(\mathbf{f})\right) \tag{8}$$

Spectral fall-off and the time t of the random walk are the main factors contributing to dimensionality reduction. Consequently, for large value of t, we will be able to capture large-scale structures in the data with fewer diffusion coordinates [3]. A natural consequence of diffusion maps based embedding is that it captures the intrinsic dimensionality of the high dimensional data which could be depicted by the way the objects are organised in the new, lower-dimensional subspace.

If semantic meaning can be associated with the intrinsic shape parameters, $\Psi^{(t)}$ can be regarded as a mapping from a low-level (but high-dimensional) shape feature space to a high-level (low-dimensional) semantic space with the abstraction level of the latter going up as t increases.

4 Experimental Results and Discussion

To evaluate the clustering performance of the proposed shape manifold learning algorithm, six shape classes were selected from the Kimia database of silhouettes [5]. These classes are: *Dog*, *Rat*, *Horse*, *Hand*, *Fish*, and *Carriage* with 49, 20, 20, 16, 32 and 20 members, respectively, making a total of 157 samples. Six randomly chosen members from each of the six classes are shown in Figure 3. To study the noise sensitivity of different algorithms, the extracted boundaries of objects were corrupted by zero mean Gaussian

noise with four different noise levels following the noise addition method outlined in Section 2.

The Fourier descriptor (**FD**) feature vector for each shape contour was used to test the clustering efficiency of diffusion maps (**DM**) against raw **FD** features and **PCA**, which can be thought of as emulating the active shape model [4] (no Procrustes analysis is required here, as the **FD** features are scale and rotation invariant). The **FD** feature vectors (with a total of 199 elements for $N = 400$ boundary points) were then used as input to the diffusion maps based embedding algorithm as well as to the classical subspace projection method of **PCA**. Figure 4 shows the two-dimensional (2D) scatter plots for the three methods at $\sigma = 0$ and $\sigma = 10$; results for $\sigma = 2,5$ are not included due to the limited space, see Table 1 and Figure 5 for quantitative results. It can be observed from Figure 4 that the first two Fourier descriptors were unable to efficiently separate all the classes except *Carriage*, *Hand*, and *Rat*. Members of the *Dog*, *Horse*, and *Fish* classes are mapped close to each other in almost all the methods. It is clear from these results that the embedding with diffusion coordinates clearly separates the former three clusters (far apart from each other) as well as better separating the latter three (which can be observed by zooming into the crowded part of the embedding using **DM**). In other words, the **DM** approach yields a higher increase in the between-class variance and reduces the within-class variance. It can also be observed from these results that noise has little effect on the clustering performance of first two Fourier descriptors as it affects the high frequency Fourier descriptors only. Interestingly, the performance of **FD** goes marginally up in the presence of noise, which can be put down to statistical variations in the feature space due to noise as well as the use of only low frequency coefficients (first and second). The **PCA** and **DM** based embeddings show signs of deterioration in their separability of clusters due to the use of high frequency coefficients which are more susceptible to noise than the low frequency coefficients.

In order to quantify the performance of the proposed shape clustering algorithm, we have used a measure known as the *class separability index* (CSI) c_i, i being the class index, defined as

$$c_i = \frac{\bar{d}_i}{\sqrt{\bar{\sigma}_i^2}} \tag{9}$$

where $\bar{d}_i$ is the average distance from the ith cluster's centroid to the centroids of all other clusters and $\bar{\sigma}_i^2$ can be regarded as a measure of its average spread estimated by the average of variances along the cluster's principal axes. Table 1 presents results of clustering performance in terms of the CSI. Also shown in Table 1 are $\bar{c}$, which is a measure of the overall clustering performance for a particular method, and $\bar{c}_i$, which can be regarded as a measure of the complexity of the ith shape and the within-class variation for its corresponding cluster. It is evident from these results that **DM** is clearly superior to **PCA** and raw **FD** at all noise levels. As the noise level increases, the overall clustering performance goes down for both **PCA** and **DM**, although $\bar{c}$ for **DM** at $\sigma = 10$ is still higher than that for the other two methods.

Application to Detection of Nuclei in Histological Images: We also tested the proposed shape clustering algorithm on images of histological tissue samples from the prostate gland. The shapes of nuclei provide clues for both the diagnosis and prognosis of cancer using computer-assisted systems aimed at quantifying changes in the nuclei shapes. To this end, the reliable isolation of nuclei from the tissue section populated with other

	c_i	$\sigma = 0$			$\sigma = 5$			$\sigma = 10$			
		FD	**PCA**	**DM**	**FD**	**PCA**	**DM**	**FD**	**PCA**	**DM**	
Dog	c_1	3.0	2.4	**12.5**	3.2	2.5	**6.7**	3.4	2.8	**7.9**	$\bar{c_1} = 4.9$
Rat	c_2	13.1	20.6	**110.8**	14.0	19.0	**76.6**	15.7	18.7	**80.8**	$\bar{c_2} = 41.1$
Horse	c_3	2.8	3.7	**8.6**	2.9	3.6	**9.1**	3.0	3.5	**5.9**	$\bar{c_3} = 4.8$
Hand	c_4	16.8	23.1	**213.1**	16.3	19.2	**72.2**	17.2	16.6	**60.2**	$\bar{c_4} = 50.5$
Fish	c_5	2.8	3.4	**15.8**	2.8	3.4	**10.8**	3.0	4.0	**9.9**	$\bar{c_5} = 6.5$
Carriage	c_6	11.4	12.9	**17.9**	11.5	12.8	**66.5**	11.7	12.5	**68.4**	$\bar{c_6} = 31.0$
	$\bar{c}$	8.3	11.0	**72.0**	8.5	10.2	**40.3**	9.0	9.7	**38.9**	

Table 1: Class separability index c_i, for $i = 1, 2, \ldots, 6$, using first two Fourier descriptors (**FD**), top two PCA coordinates (**PCA**), and top two (2nd and 3rd) diffusion coordinates (**DM**), for different levels σ of noise.

undesirable objects is pivotal for the success of these systems. Figure 6(*a*) shows a typical tissue section from prostate gland. The tissue section images were binarised using a intensity-based *k*-mean clustering scheme. Diffusion maps for the **FD** feature vectors of the various object boundaries were computed. The potential nuclei identified by a simple zero-crossing applied to the top two coordinates in the embedding are shown in Figure 6(*b*). From these preliminary results, it can be observed that our algorithm has successfully detected all regular-shaped nuclei (shown in *green*), effectively separating them from the remaining objects, such as parts of nuclei, overlapping nuclei clusters, nuclear debris, and extracellular objects (shown in *red*). One of the major advantages of this detection scheme is its low computational complexity. It only needs two thresholding operations to separate regular-shaped nuclei from other objects. The spatial arrangement of these nuclei as dictated by the diffusion mapping can be semantically interpreted as well (see [1] for more details).

5 Conclusions

Extraction of the intrinsic parameters of shape is of vital importance for a wide range of applications. In this paper, we have presented a novel framework for learning the global structure of shape manifolds using diffusion maps in an unsupervised manner. The clustering results using the proposed framework show its superiority over classical shape clustering using linear dimensionality reduction. The difficulty in separating members of *Dog* and *Horse* classes stems from the fact that their contours have certain subtle differences which could not be picked up by the first two diffusion coordinates. It is our observation that the class separability indices for these two classes can be significantly increased by increasing the number of diffusion coordinates. A naïve extension of the proposed shape clustering algorithm for the shape classification problem would require re-computing the Markov matrix every time there is a new test sample. This extra computation can be avoided by employing out-of-sample extension methods (such as the Nyström formula) [16] to embed a new sample into the low-dimensional diffusion coordinate space without having to re-compute the Markov matrix. The proposed framework can also be extended for it to be applied to the general problem of shape modelling.

References

[1] M. Arif and N.M. Rajpoot. Detection of nuclei by unsupervised manifold learning. In *Proceedings Medical Image Understanding and Analysis (MIUA)*, July 2007.

[2] G. Charpiat, O. Faugeras, and R. Keriven. Approximations of shape metrics and application to shapewarping and empirical shape statistics. *Foundations of Computational Mathematics*, 2005.

[3] R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis, Special Issue on Diffusion Maps and Wavelets*, 21:5–30, July 2006.

[4] T. Cootes, D. Cooper, C. Taylor, and J. Graham. Active shape models – their training and applications. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[5] D.Sharvit, J.Chan, H.Tek, and B.B.Kimia. Symmetry-based indexing of image databases. *Journal of Visual Communication and Image Representation*, 1998.

[6] P. Etyngier, R. Keriven, and J.-P. Pons. Towards segmentation based on a shape prior manifold. In *1st International Conference on Scale Space andVariational Methods in Computer Vision (SSVM)*, 2007.

[7] X. He, W.-Y. Ma, and H.-J. Zhang. Learning an image manifold for retrieval. In *Proceedings ACM International Conference on Multimedia*, pages 17–23, 2004.

[8] H. Kauppinen, T. Seppanen, and M. Pietikainen. An experimental comparison of autoregressive and fourier-based descriptors in 2-d shape classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:201–207, 1995.

[9] D.G. Kendall. Shape manifolds, procrustean metrics, and complex projective spaces. *Bulletin of the London Mathematical Society*, 16(2):81–121, 1984.

[10] S. Lafon and A.B. Lee. Diffusion maps and coarse-graining: A unified framework fordimensionality reduction, graph partitioning, and data setparmeterization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9), September 2006.

[11] Y. Lu, S. Schlosser, and M. Janeezko. Fourier descriptors and handwritten digit recognition. *Machine Vision and Applications*, 6:25–34, 1993.

[12] O. Faugeras M. Leventon, E. Grimson. Statistical shape influence in geodesic active contours. In *Proceedings International Conference onComputer Vision and Pattern Recognition (CVPR)*, 2000.

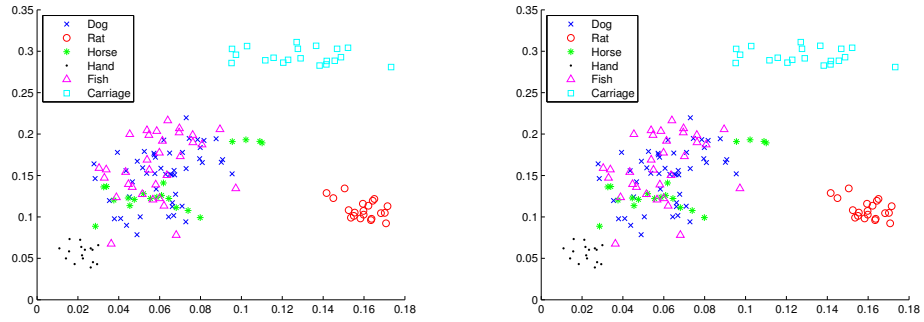
[13] A. Sekuler. Local and global minima in visual completion:effects of symmetry and orientation. *Perception*, 23:529–545, 1994.

[14] R. Souvenir and R. Pless. Manifold clustering. In *Proceedings International Conference on Computer Vision (ICCV)*, pages 648–653, 2005.

[15] C.J. Twining and C.J. Taylor. Kernel principal component analysis and the construction of non-linear active shape models. In *Proceedings British Machine Vision Conference (BMVC)*, 2001.

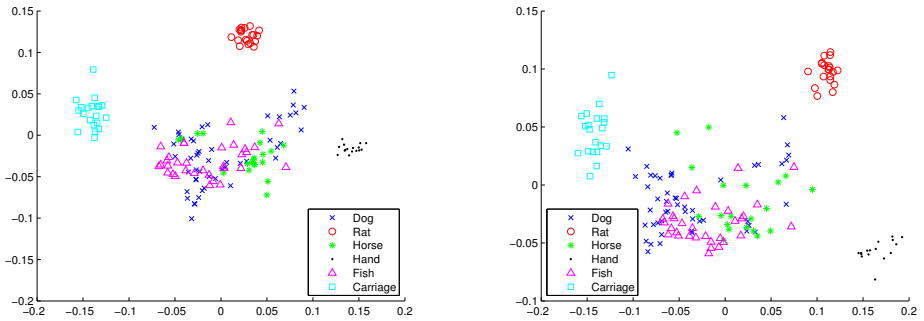
[16] L.J.P. van der Maaten, E.O. Postma, and H.J. van den Herik. Dimensionality reduction: A comparative review. Preprint, 2007.

[17] D. Yankov and E. Keogh. Manifold clustering of shapes. In *Proceedings International Conferenceon Data Mining*, 2006.



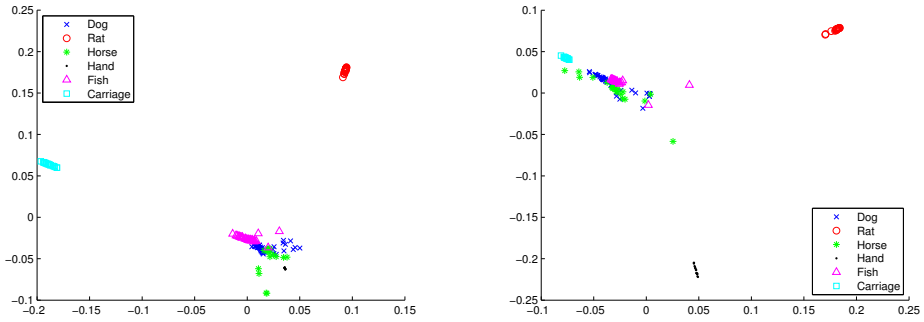
$\sigma = 0$ $\sigma = 10$

Two-dimensional feature space using top two (1st and 2nd) Fourier descriptors



$\sigma = 0$ $\sigma = 10$

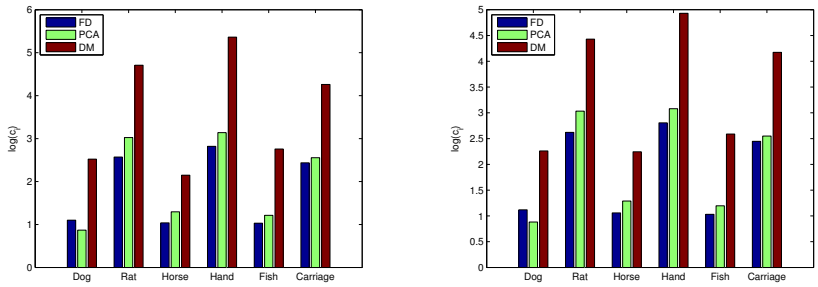
Scatter plots of top two PCA coordinates



$\sigma = 0$ $\sigma = 10$

Scatter plots of top two diffusion coordinates

Figure 4: Two-dimensional feature space (*Top*) and 2D embeddings using PCA (*Middle*) and diffusion maps (*Bottom*).



(a) $\sigma = 0$ (b) $\sigma = 2$

(c) $\sigma = 5$ (d) $\sigma = 10$

Figure 5: *Clockwise from Top-Left*: Logarithm of the class separability index c_i, for $i \in \{Dog, Rat, Horse, Hand, Fish, Carriage\}$, using top two Fourier descriptors (**FD**), PCA coordinates (**PCA**), and diffusion coordinates (**DM**) for $\sigma = 0,\ 2,\ 5$, and 10.

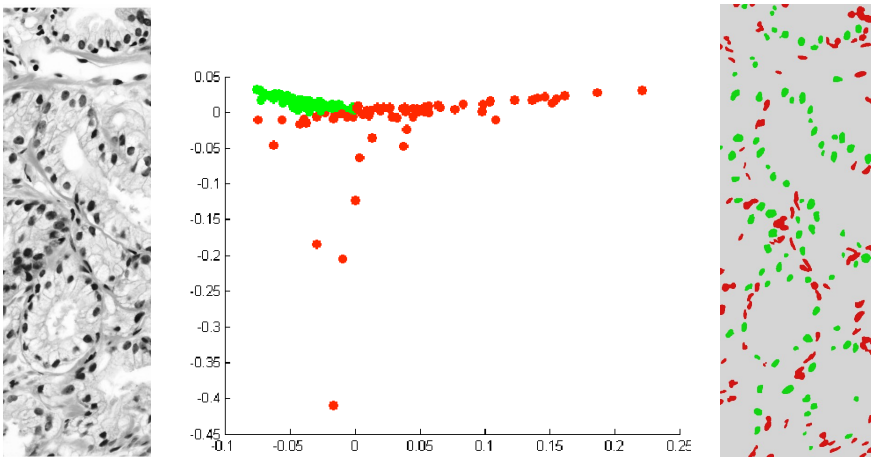


Figure 6: Detection of nuclei: A prostate histology image (*Left*), 2D embedding of the feature vectors for nuclei contours (*Middle*), and Regular-shaped nuclei (shown in green colour) detected by zero-crossings of the diffusion coordinates (*Right*).

Finsler Level Set Segmentation for Imagery in Oriented Domains

Vandana Mohan, John Melonakos and Allen Tannenbaum
Electrical and Computer Engineering,
Georgia Institute of Technology, Atlanta GA, USA

Marc Niethammer
Surgical Planning Laboratory,
Brigham and Women's Hospital, Boston MA, USA

Marek Kubicki
Laboratory of Mathematics in Imaging,
Brigham and Women's Hospital, Boston MA, USA
vandana.mohan@gatech.edu

Abstract

In this paper, we present a novel directional level set segmentation framework employing the theory of Finsler active contours. The framework provides a natural way to perform segmentation of image data in oriented domains. We share examples of this technique on diffusion-weighted magnetic resonance imagery (DW-MRI) for the segmentation of neural fiber bundles and we show examples of texture based segmentation using structure tensors. We also demonstrate that for some applications higher accuracy is achieved by the proposed framework than by level set methods that employ Riemannian metrics. This gain is attributed to the relaxation of the tensor model constraint which is imposed upon the metric in the Riemannian case.

1 Introduction

Image segmentation is the problem of separating a given image into areas of interest, based on image-derived features. These features could include intensity and edge information, regional statistics, etc. In many scenarios, it is beneficial to transform the image into a space where the features of interest can be better discriminated. One such instance is when intensity information is insufficient to segment an image. In this case, it is desirable to use additional features to help discriminate the object of interest from the background (such as using structure tensors to capture textural information). Interesting approaches in this vein include recent work on the application of structure tensors for image segmentation [2]. The structure tensor is a feature that essentially converts the given isotropic image data into an oriented domain. In this work, we present a method which is able to segment features of interest in oriented domains of arbitrary dimension via the level set method. In addition to the structure tensor example, there are other naturally occurring instances of

imagery in oriented domains - for example, diffusion-weighted magnetic resonance imagery (DW-MRI). In this work, we build upon our previous work based on Finsler active contours [3]. Specifically, we report the first formulation of this Finsler level set method in arbitrary dimension and the first application of the framework to a variety of oriented domains minimizing novel energy-based direction-dependent functionals.

The segmentation problem in oriented domains is interesting because the image data is a function of not only position but also direction. Since the segmentation of DW-MRI data is inherently a directional problem, and texture segmentation can be recast as the segmentation of an oriented image, it is expected that a segmentation framework which naturally incorporates directionality - like the Finsler active contour framework - can be applied for both in fairly similar ways.

There has been a great deal of research in the direction of finding and characterizing neural connections between brain structures in diffusion-weighted magnetic resonance imagery. The cingulum bundle (CB) is a structure in the brain that has recently become the subject of interest as an anatomical structure which might display quantifiable differences between schizophrenic and normal control populations and thus aid towards the diagnosis of this disorder. The cingulum bundle is a fiber bundle and thus made up of several individual fibers, mostly disjoint and potentially intersecting at points, all roughly similarly aligned. Thus the fiber bundle clearly possesses distinct directionality from the surrounding areas of the brain, which could be either intersecting fibers running roughly perpendicular to the CB, or structures with strong but isotropic diffusion profiles like the cerebral spinal fluid (CSF).

It should be noted that segmentation using a Riemannian approach can work with tensor data (and thus handle the structure tensors), however it enforces an elliptic constraint on the data, by the very nature of approximation of these tensors. There are usage scenarios where the data at hand might not be accurately captured by a tensor model. Thus, extracting textural information as a tensor is an example of a situation where generality in the nature of this feature might allow for better representation and thus better segmentation. The Finsler active contour framework presented in this paper offers a natural way to work with such general directional features.

In this paper, we extend the results of [3] by presenting the first formulation and results of a level set implementation of this framework in arbitrary dimension. We have successfully applied this technique to DW-MRI data and tested its application in the case of image segmentation using textural information. We also illustrate the advantage of the framework in its ability to handle directional data that is not expressed (or approximated) to fit a tensor model, and the increase of accuracy of the segmentation obtained as a result.

1.1 Textural Segmentation

The problem of texture-based segmentation has been explored extensively in [2] and [5] and the references there-in. The motivation for the application of this framework to textural segmentation is to exploit the directionality of the feature - the structure tensor - being extracted. As opposed to current methods that employ regional statistics on the structure tensor space, this work finds edges in this oriented space. Thus, it can also easily be extended to images of any higher dimension. As shall be discussed subsequently in this paper, the Finsler-based framework removes the need to fit a tensor model to the data, and can thus be expected to better discriminate features of interest.

1.2 DW-MRI Segmentation - Identification of the Cingulum Bundle

The cingulum bundle is a 5-7 mm in diameter fiber bundle that interconnects all parts of the limbic system. It originates within the white matter of the temporal pole, and runs posterior and superior into the parietal lobe, then turns, forming a "ring-like belt" around the corpus callosum, into the frontal lobe, terminating anterior and inferior to the genu of the corpus callosum in the orbital-frontal cortex [6]. Moreover, the CB consists of long, association fibers that directly connect temporal and frontal lobes, as well as shorter fibers radiating into their own gyri. The CB also includes most afferent and efferent cortical connections of cingulate cortex, including those of prefrontal, parietal and temporal areas, and the thalamostriatae bundle. In addition, lesion studies document a variety of neurobehavioral deficits resulting from a lesion located in this area, including akinetic mutism, apathy, transient motor aphasia, emotional disturbances, attentional deficits, motor activation, and memory deficits. Because of its involvement in executive control and emotional processing, the CB has been investigated in several clinical populations, including depression and schizophrenia. Previous studies, using DTI, in schizophrenia, demonstrated decrease of FA in anterior part of the cingulum bundle [1, 8], at the same time pointing to the technical limitations restricting these investigations from following the entire fiber tract.

Given the directional nature of the cingulum bundle, it is of interest to test the efficacy of the proposed framework in automatically segmenting this structure from a given 3D brain volume.

2 Background

2.1 Motivation

For image data in oriented domains, the boundaries of objects of interest extremize energy functionals that depend not only on position but also on direction. This is the fundamental motivation behind the Finsler active contour framework since it attempts to extremize an energy of exactly this form. Prior work has implemented the dynamic programming based approach to this framework. The work documented in this paper documents the first level set implementation of this framework in arbitrary dimension and the results obtained with this implementation for 2D and 3D images. This implementation has the natural advantages of being able to handle topological changes, and being able to trivially extend our implementation to arbitrary dimension. Also, it has the advantage that it can accommodate a variety of cost functions since it accepts the cost function as a scalar field and thus the actual implementation requires no alterations from the basic setup.

2.2 Theory and math

Let $\Psi(p,d)$ denote the energy that we desire to minimize as a function of the position 'p' and the direction 'd'. The expression for the evolution of the hypersurface is obtained as,

$$\Sigma_t = -\{\nabla_p \Psi.N + Tr(\nabla_{dd}\Psi) + \Psi.Tr(dN)\}N, \tag{1}$$

where Σ denotes the hypersurface, Ψ is the local cost, N denotes the unit normal to the hypersurface, dN is the shape operator. The detailed derivation of this expression can be

found in [4].

Denoting the surface of interest as the zero level set of a signed distance function, we obtain the expression for the evolution of the level set function as,

$$u_t = \nabla_p \Psi . \nabla u + \{ Tr(\nabla_{dd} \Psi) + \Psi . Tr(dN) \} \parallel u \parallel, \tag{2}$$

where u is the level set function.

This expression can be separated into the terms that influence diffusion and terms that are hyperbolic in nature. In the numeric implementation of this framework, appropriate differencing methods are employed for the individual terms depending upon which category they belong to.

3 Implementation

The implementation of the Finsler active framework documented in this paper is done in arbitrary dimension. The terms from the expression for the hypersurface evolution that are challenging to compute in arbitrary dimension are the shape operator, dN, and surface derivatives at each point on the on the hypersurface. In this work, the level set approach of [7] is used to implement this framework and by an appropriate choice of the level set function as a signed distance function, it is possible to obtain simplified expressions for these two terms as discussed in subsequent sections.

3.1 Level Set Implementation

The level set formulation implicitly represents the surface of interest as the zero level set of an appropriately defined level set function. This work uses the signed distance function because of its desirable properties with regards to the computation of geometrical quantities of interest in surface evolution. This choice simplifies the computation of the Shape operator, dN, and a basis for the tangent space (which is equivalent to the surface derivatives with respect to the arc length parameter s).

3.2 Shape operator

Let u denote the signed distance function (SDF). Then, the normal can be obtained from the gradient of the SDF as,

$$N = \frac{\nabla u}{\parallel \nabla u \parallel}. \tag{3}$$

Since the shape operator is the derivative of the normal to the surface with respect to arc length, s, we can write it as,

$$\frac{dN}{ds} = \frac{d}{ds} \frac{\nabla u}{\parallel \nabla u \parallel}. \tag{4}$$

This can be simplified to obtain an expression for the shape operator in terms of the SDF as

$$dN = \frac{1}{\parallel \nabla u \parallel} (I - N.N^t) H_u. \tag{5}$$

where H_u is the pointwise hessian of the levelset function 'u'.

3.3 Computation of surface derivatives

It can be seen that that with arc length parameterization and the fact that the surface derivatives required in the numeric implementation occur under a trace operator (which is invariant under a change of basis of the matrix argument or a change of order of arguments), we can replace the matrix containing surface derivatives by an arbitrary choice of basis for the tangent space at the point. This basis is trivial to determine if the unit normal to the surface is known. Since for an SDF, the normal can be obtained from the gradient, we can simply express the matrix with the tangent space basis as,

$$d\Sigma = (I - N.N^t), \tag{6}$$

where $d\Sigma$ denotes the surface derivatives used in the evolution, and N denotes the unit normal to the surface Σ.

It can be shown that the eigenvectors of this matrix are in fact the normal itself - corresponding to zero eigenvalue - and the (n-1) tangent directions corresponding to eigenvalue 1 which represent surface derivatives.

The sparse field approach of [9] was chosen to implement the level set formulation of the Finsler active contour framework, in order to achieve fast computations given the large sizes of the directional datasets being used.

3.4 Cost functions

The fundamental motivation to all the cost functions explored in this work is to find edges in the directionality of the underlying data. Hence all the cost functions discussed here yield segmentations that are primarily edge-based in nature. A crucial feature of the available data is that the object of interest does not homogeneously display a single direction. There are intermediate areas where (using DW-MRI terminology), the diffusion is possibly isotropic and sometimes zero. Thus, it is important to suitably compute gradients in these images to better capture the directionality inside (and outside) the evolving surface. The cost functions are formulated in such a way that the costs ranges from 0 to 1, and so that the energy is minimized on the surface of interest. The convergence condition was fairly simple with the algorithm stopping when the average cost on the surface fell below 0.1 and stayed there beyond a certain number of iterations.

4 Experiments and Results

The framework was tested via three sets of experiments. The first aimed to compare it with the Riemannian approach on a tensor fit of the data. The second experiment applied the framework to the segmentation of a texture image and the third applied it towards automatically finding the cingulum bundle from DW-MRI imagery.

4.1 Synthetic data

One of the advantages of the Finsler active contour framework is the fact that the data does not need to be made to fit a tensor model. This allows greater accuracy in capturing features of interest. This is demonstrated via a comparison between the proposed framework and a Riemannian approach applied to a tensor approximation of the same data set.

This data set is synthetic with different diffusion profiles within and outside the object of interest which is a simple circle. It is an extreme example since the diffusion profiles within and outside the circle look identical when fit to a tensor model (Table 1, and the Riemannian approach eventually causes the initial surface to just collapse to a point under the influence of curvature, since under the tensor model, the energy landscape is flat and 1 (maximum) everywhere! This experiment though simplified, clearly illustrates the advantage of our framework.

4.2 Texture Segmentation

For textural segmentation, the transformation of the available data into an oriented domain is crucial to the effectiveness of the segmentation algorithm. This transformation is dependent upon the nature of the image that we desire to segment. In the images used in this work, it is of interest to separate the directionality of the gradients of the image inside and outside the object. Hence, the transformation that we employ to construct the feature data for this application is to compute the directional gradients of the given image. The gradients are comparable to the diffusion profile for DW-MRI imagery and the directions in which these gradients are computed are analogous to the sampling directions. Figure 2 illustrates this. Figure 3 shows a comparison of the method with the intensity edge-based approach, and clearly demonstrates the advantage we gain by adding directionality into the segmentation problem via the transformation into an oriented domain. It is observed that with a purely intensity edge-based approach, for this image, the segmentation process is not well-behaved and does not converge.

4.3 DW-MRI data

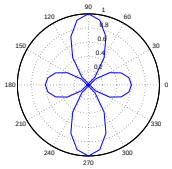
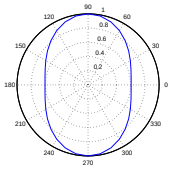
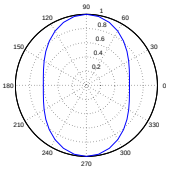
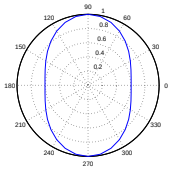
The object of interest in the available DW-MRI imagery is the cingulum bundle. We can deduce the equivalence between such a dataset and the image on which the textural segmentation has been demonstrated. The 'stripes' in the textural image are in a sense equivalent to the fibers within the cingulum bundle that we desire to encapsulate within a single surface. The same cost function was in fact applied to both these data sets with no difference other than the dimension of the data itself. The initial level set for this application was arrived at by creating the surface of smallest volume that fully contains the anchor tract that is obtained out of the Finsler tractography module discussed in [3].

In DW-MRI data, the task of segmenting the cingulum bundle is equivalent to find disjoint fibers in the available brain volume that are similarly aligned. In the absence of ground truth for the cingulum bundle in the available data, we use the availability of simple eigenvalue analysis based approximations to the cingulum bundle along with visual verification to validate the performance of the algorithm on the DW-MRI data. Based on these, the CB extracted is found to be satisfactory as shown in figure 4.

5 Conclusions and Future Work

In this work, we have introduced the Finsler level set framework which is a novel way to perform segmentation of imagery in oriented domains. We demonstrate the advantage of the framework over Riemannian approaches that impose a tensor model on the data. We have also shown the application of this framework in segmenting the cingulum bundle

Table 1: Diffusion profiles in the synthetic image. Results of comparison in Figure 1

	Diffusion profile inside	Diffusion profile outside
Original data		
Tensor model approximation		

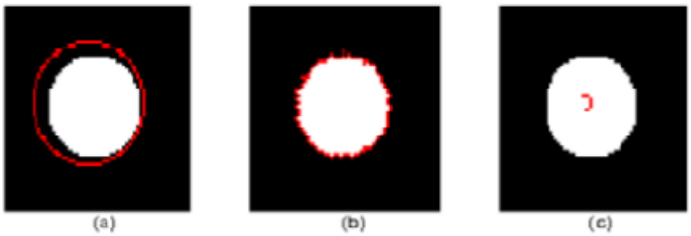


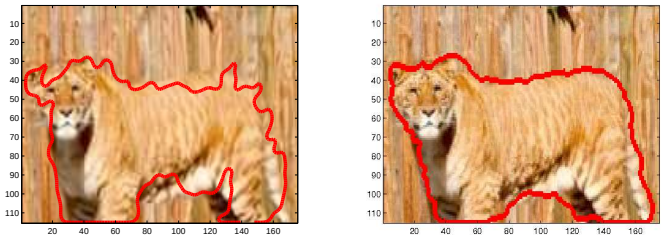
Figure 1: Results on synthetic data comparing Finsler-based framework to Riemannian approach using tensors. (a) shows the initial level set. (b) shows the result obtained using the Finsler level set framework, and (c) shows the result of applying Riemannian level sets on a tensor fit to the data. (Diffusion profiles are provided in Table 1)

from DW-MRI imagery and a proof-of-concept of its utilization in texture segmentation, while simultaneously drawing the parallels between the two cases that allow us to use similar approaches. The algorithm is found to perform well in both these cases.

The framework gives us a method to automatically segment the cingulum bundle from the DW-MRI volume. The results have been validated by comparison with eigen value based thresholding and visual inspection. Further validation will be explored by incorporating the algorithm into a system that can automatically extract the CB and perform statistical analysis on the CB towards differentiating normal controls from schizophrenic controls. Also, the texture segmentation examples shown in this paper have been chosen because of their similarity to the nature of the DW-MRI data in the sense of separating directionality inside and outside the object, and to demonstrate the advantage of adding directionality to the image segmentation framework. Future work will attempt to apply this framework to more texture images. Also, since the costs employed in this work are

(a) (b)

Figure 2: Result on texture image. (a) shows the initial levelset and (b) shows the (smoothed) result from the Finsler levelset framework



(a) (b)

Figure 3: Comparison between (a) intensity edge-based approach and (b) directional edge-based (proposed) approach for same initial level set.

purely edge-based and hence we face natural limitations on the placement of the initial levelset, it is also of great interest to explore costs of a more regional nature and evaluate their effectiveness in texture segmentation as well as on the DW-MRI data.

References

[1] M. Kubicki, C.F. Westin, P.G. Nestor, C.G. Wible, M. Frumin, S.E. Maier, R. Kikinis, F.A. Jolesz, R.W. McCarley, and M.E. Shenton. Cingulate fasciculus integrity disruption in schizophrenia: a magnetic resonance diffusion tensor imaging study. *Biological Psychiatry*, 54(11):1171–1180, 2003.

[2] J. Malcolm, Y. Rathi, and A. Tannenbaum. A graph cut approach to image segmentation in tensor space. In *Workshop on Component Analysis Methods (CVPR)*, 2007.

[3] J. Melonakos, E. Pichon, S. Angenent, and A. Tannenbaum. Finsler Active Contours. *To appear in IEEE Trans. PAMI*, 2007.

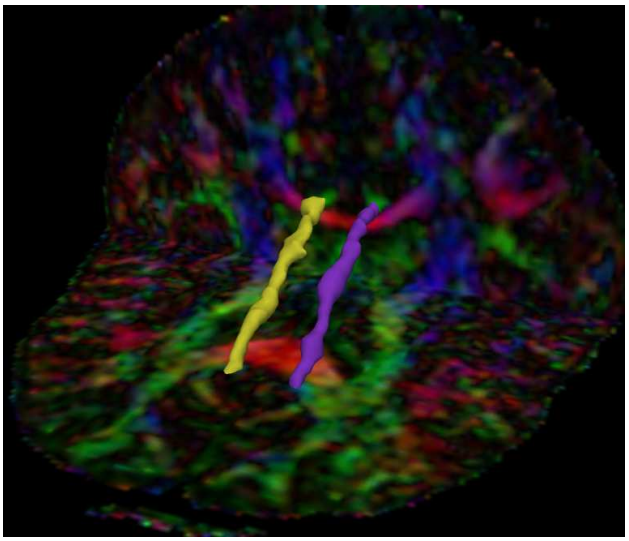


Figure 4: Result on DW-MRI data. The structures in yellow and magenta indicate the CB surface identified in the right and left hemisphere of the brain volume.

[4] E. Pichon. *Novel methods for multidimensional image segmentation*. PhD thesis, Georgia Institute of Technology, 2005.

[5] M. Rousson, T. Brox, R. Deriche, I. Projet Odyssee, and F. Sophia-Antipolis. Active unsupervised texture segmentation on a diffusion based feature space. *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, 2, 2003.

[6] J.D. Schmahmann and D.N. Pandya. *Fiber Pathways of the Brain*. Oxford University Press, 2006.

[7] J.A. Sethian. *Level set methods and fast marching methods*. Cambridge University Press Cambridge, 1999.

[8] F. Wang, Z. Sun, L. Cui, X. Du, X. Wang, H. Zhang, Z. Cong, N. Hong, and D. Zhang. Anterior Cingulum Abnormalities in Male Patients With Schizophrenia Determined Through Diffusion Tensor Imaging, 2004.

[9] R.T. Whitaker. A Level-Set Approach to 3D Reconstruction from Range Data. *International Journal of Computer Vision*, 29(3):203–231, 1998.

Automated Segmentation of Low Light Level Imagery using Poisson MAP-MRF Labelling

Hugh Gribben[1], Paul Miller[1], Hongbin Wang[1], and Mark Browne[2]

[1]The Institute of Electronics, Communications and Information Technology (ECIT)
Queen's University Belfast, Belfast, UK. BT3 9DT

[2]Andor Bioimaging Division, Morrisville, USA. NC 27560

hgribben01@qub.ac.uk

Abstract

An automated unsupervised technique, based upon a Bayesian framework, for the segmentation of low light level imagery is proposed. Primarily, Mixture Modelling is used to provide a baseline estimate. This estimate is then refined to consider spatial correlations using Markov Random Field (MRF) Modelling. The technique has been implemented assuming low-light level Poisson statistics, and the results compared to the more widely used assumption of Gaussian statistics. Investigations revealed the Poisson technique quantitatively outperforms the Gaussian technique for synthetic low light imagery, both before and after avalanche multiplication, via the multiplication register of a Low-light Level Charge Coupled Device (L3CCD). The technique was then applied to the task of segmenting a biomedical dataset obtained from a L3CCD. Qualitative results were promising, again showing improvement over the Gaussian technique.

1 Introduction

The high sensitivity of recently developed L3CCD sensors provides us with the ability to acquire multidimensional datasets at high temporal sampling rates under low-light level conditions. This allows, for the first time, the study of the dynamics of biomedical bodies such as cells, and their intra-cellular components. The novel imaging method allows vast multidimensional datasets to be regularly acquired. Novel statistical distributions are present within the data; due to both the initial low-light levels at which the data is acquired, and the avalanche multiplication to which the data is subjected via the L3CCD. Along with the size of these datasets, the statistical distributions present a new challenge to improve upon the existing methods currently in use within the area of biomedical image processing. In this work, an automated unsupervised technique, based upon the Bayesian framework and assuming low-light level statistics, is proposed for the segmentation of L3CCD imagery.

Related work in this area has included the segmentation of medical images using an automated volumetric segmentation system [1]. In this work, the authors combine techniques in the areas of statistical reasoning and multi-resolution analysis, to differentiate tissue types in brain images obtained from Magnetic Resonance Imaging (MRI) and Positron Emission Tomography (PET) imaging techniques. Similarly to Peng et. al. [2], in which a method is presented to segment 3D MRI brain images, a Spatial Gaussian Mixture Model is used to represent the intensity probability distribution of each of the brain tissues. Like the work presented here, this method consists firstly of a learning process based on the Expectation Maximisation (EM) algorithm to estimate parameters. A classification algorithm based upon the Iterated Conditional Modes (ICM) algorithm is then used to perform segmentation of the biomedical imagery; using these parameters. In both cases, the results found show an accurate method for the segmentation of the example images. Deng [3], using a similar method for unsupervised MRF modelling, has also found encouraging results. While related works [1-4] have adopted a Gaussian Maximum a Posterior-MRF (MAP-MRF) approach, we instead assume low-light level Poisson statistics for the optimisation of the novel L3CCD data; and apply the resulting technique. To the authors' knowledge, this Poisson MAP-MRF approach has not previously been applied to this class of imagery. The paper is organised as follows. In section 2, our Poisson MAP-MRF approach is defined. In section 3, results obtained are reviewed and analysed. Finally, conclusions and future work are discussed in section 4.

2 Poisson MAP-MRF Approach

2.1 Low Light Level Statistics and L3CCD Theory

At low light levels (up to approx. 50 photons per pixel per integration time), photon counts are distributed according to the Poisson distribution [5]

$$p(y \mid \mu) = \exp\big(y \log(\mu) - \mu - \log(y!)\big) \tag{1}$$

where y is a non-negative number of occurrences, and μ the mean number of occurrences. L3CCDs contain a multiplication register which implements electron avalanche multiplication so that a large mean gain can be realised prior to the conventional readout amplifier. When the photon input level is small and the gain large, the L3CCD output can be estimated by providing the probability distribution of the L3CCD with (1), giving the joint distribution [5]

$$p(x,u) = \sum_{n=1}^{\infty} \frac{\exp\left(-\mu - x/g\right)\mu^n \left(x/g\right)^{n-1}}{g\,(n-1)!\,n!} \tag{2}$$

This is the probability the output will be x when the mean light level is μ, with the input number of photons equal to n, and a mean gain of g.

2.2 Derivation of Poisson MAP-MRF Technique

Using Bayes estimation, a risk is minimized to obtain an optimal estimate. In terms of the segmentation problem, according to the Bayes rule, the posterior probability of a labelling estimate X given an observation Y can be computed by $P(X|Y) = P(Y|X)P(X)/P(Y)$ where $P(X)$ is the prior probability of X, $P(Y|X)$ is the likelihood function of X with respect to Y, and $P(Y)$ is the density of Y. The minimal risk estimate is therefore $X^* = \arg\max_X P(X|Y)$, the MAP estimate. As $P(Y)$ is a constant for fixed Y, the MAP estimate is equivalently found by

$$X^* = \arg\max_X \{P(Y|X)P(X)\} \tag{3}$$

In many applications, an initial estimate for X is obtained using mixture modelling, assuming a mixture density of Gaussian distributions. However, taking into account the Poisson nature of the low-light level imagery we wish to segment, we model the intensity distribution as a mixture of Poisson distributions, given by

$$f(Y_i \mid \theta, \lambda) = \sum_{k=1}^{K} \lambda_k f_k(Y_i \mid \theta_k) \tag{4}$$

where K is the number of assumed Poisson distributions, Y_i = observed value at pixel i, θ_k = the set of mean vectors $\{\mu_k\}$; $f_k(\cdot \mid \theta_k)$ is a Poisson density with mean μ_k; $\theta = (\theta_1, \ldots, \theta_k)$ and $\lambda = (\lambda_1, \ldots, \lambda_K)$ is a vector of mixture probabilities such that $\lambda_k \geq 0 \ (k = 1, \ldots, K)$ and $\sum_{k=1}^{K} \lambda_k = 1$ [1].

The algorithm used in practice to find the mixture of distributions that best model the dataset is the EM algorithm, first introduced by Dempster et al. [6]. This is an iterative algorithm, which estimates the parameters via a Maximum Likelihood (ML) criterion. The mixture density returned can then be used to associate pixel observations with a Poisson density k using a simple ML estimation

$$k^* = \arg\max_k f_k(Y_i \mid \theta_k) \tag{5}$$

A label X_i is then given to pixel i in the estimate X, corresponding to k^*.

In MAP-MRF labeling, $P(X|Y)$ is the posterior distribution of a MRF. The prior model $P(X)$ takes into account spatial correlations present in an image, and is dependent upon the type of scene. Assuming our scene to be a piecewise constant surface, we consider an indicator function, $I(X_i,X_j) = 1$ if $X_i = X_j$ and otherwise = 0. The Potts model can be described by $P(X) \propto \exp\left(\phi \sum_{i,j} I(Xi, Xj)\right)$, where the sum is computed over all neighbour pairs. Spatial homogeneity in the model is expressed using the parameter ϕ, small values implying randomness, and large values implying uniformity [7]. Let $N(X_i)$ be the neighbourhood of X_i, and let $U(N(X_i),k)$ be the number of neighbourhood pixels with a label corresponding to a distribution k. The prior energy for each distribution labelled k can then be defined as the negative of the sum of all the clique potentials over X [8]

$$E(X) = -\sum_{i \in S} \phi U(N(X_i), k) \tag{6}$$

where S is the set of pixel sites, and i the site currently under consideration.

The likelihood model $P(Y|X)$ depends upon physical considerations. With the traditional approach, Y is assumed to be a degraded version of a MRF realization X due to independent additive Gaussian noise. Taking a Gaussian distribution as a special form of a Gibbs distribution, advantage is then taken of a MRF's equivalence to a Gibb's distribution [8], to then define the likelihood as $P(Y \mid X) = \exp\left(-E(Y \mid X)\right) \Big/ \prod_{i \in S} \sqrt{2\pi\sigma_i^2}$, where σ_i^2 is the variance of the estimated Gaussian distribution at pixel i, and

$$E(Y \mid X) = \sum_{i \in S} \left(X_i - Y_i\right) \Big/ 2\sigma_i^2 \tag{7}$$

is the likelihood energy. Instead, we assume low-light statistics, taking each observed pixel value to be the output of a Poisson process. We then take (1) as a special form of the Gibbs distribution, as opposed to the Gaussian distribution. Next, we take advantage of a MRF's equivalence to a Gibb's distribution to define the likelihood as $P(Y \mid X) = \exp\left(E(Y \mid X)\right)$ where

$$E(Y \mid X) = -\sum_{i \in S} \left(Y_i \log\left(X_i\right) - X_i - \log\left(Y_i\right)\right) \tag{8}$$

is the likelihood energy. Finally, the prior and likelihood energies are added to yield the posterior energy. In the Gaussian case, this gives

$$E(X \mid Y) = \sum_{i \in S} \left(X_i - Y_i\right) \Big/ 2\sigma_i^2 - \sum_{i \in S} \phi U(N(X_i), k) \tag{9}$$

For our updated Poisson technique, we replace (9) with

$$E(X \mid Y) = -\sum_{i \in S} \left(Y_i \log\left(X_i\right) - X_i - \log\left(Y_i\right)\right) - \sum_{i \in S} \phi U(N(X_i), k) \tag{10}$$

The MAP estimate can then be found by then minimising the posterior energy. Practically, this can be performed by the use of the ICM algorithm, originally introduced by Besag [9]. This is an iterative algorithm that begins with the observed scene Y, and the initial estimate of the true scene X from mixture modelling. By considering each pixel site in turn, it then proceeds to provide a new estimate of the true scene iteratively, until convergence is reached, or a maximum number of iterations complete. The optimal value of ϕ for each iteration is chosen via Pseudo-Likelihood Information Criterion (PLIC) analysis. This is an automated Bayesian technique that considers the ratio of likelihoods of output models to determine the optimal model. In the Gaussian case, a single iteration of the ICM requires (11) for each pixel i, where μ_{kG} is the Gaussian mean of state k.

$$k^* = \arg\min_k \left(\mu_{kG} - Y_i\right) \Big/ 2\sigma^2 - \phi U(N(X_i), k) \tag{11}$$

A label X_i is then given to pixel i in the updated estimate X, corresponding to k^*. For our updated Poisson technique, we replace (11) with

$$k^* = \arg\min_k - \left(Y_i \log\left(\mu_{kP}\right) - \mu_{kP} - \log\left(Y_i\right)\right) - \phi U(N(X_i), k) \tag{12}$$

where μ_{kP} is the Poisson mean of state k.

3 Results and Analysis

3.1 Segmentation of Low-Light Data

The proposed technique was initially applied to synthetic data with mean gain set to 1 for analytical purposes. In the first case, an image is produced by the creation of a binary random pattern, by convolving a normally distributed 256x256 random array with a Gaussian filter. This allows us to quantify the spatial dependencies within the data by comparison of the Full Width at Half Maximum (FWHM) parameter of the filter used (The greater the FWHM, the greater the spatial dependencies). Pixels are then labelled depending upon value. This provides us with our "true scene" $\hat{X}$ that we wish to recreate via our segmentation technique (Fig. 1(a)). Poisson noise is then added to each pixel, the mean Poisson value being dependent upon the initial labelling, to give low-light synthetic data (Fig. 1(b)). Figs. 1(c) and 1(d) show qualitative results giving the initial estimate returned from the mixture modelling step, and the application of the MAP-MRF technique. Here, Poisson noise is added to the true scene with means equivalent to light levels of 4 and 10 photons/pixel/integration time, dependent upon the initial labelling. The application of the MAP-MRF technique shows considerable improvement over the initial estimate returned from the mixture modelling step, as well as a close approximation to the true scene we are attempting to recreate.

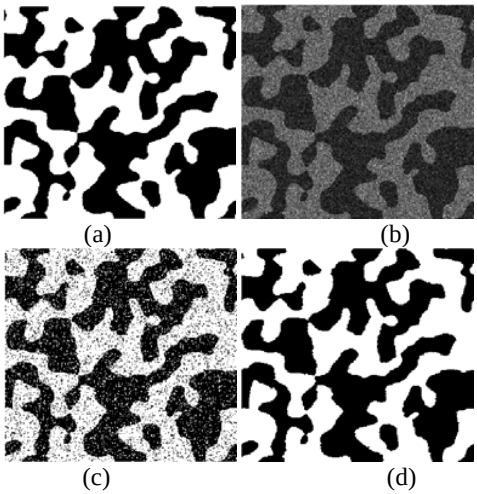


Figure 1: Automated Segmentation of Synthetic Data (a) 'True Scene' with FWHM = 11.77 (b) 'Observed Scene' (True Scene with Poisson noise added (c) Output - Mixture Modelling Segmentation of b (d) Output – Poisson MAP-MRF Segmentation of b

Fig. 2 shows quantitative results obtained from the MAP-MRF technique compared to the initial estimate returned from the mixture modelling step, as ϕ is increased over 10 iterations of the ICM algorithm for different initial true scenes with increasing spatial dependencies from 2(a) through 2(c). Segmentation was performed 10 times for each differing spatial dependency, and the average error rates calculated.

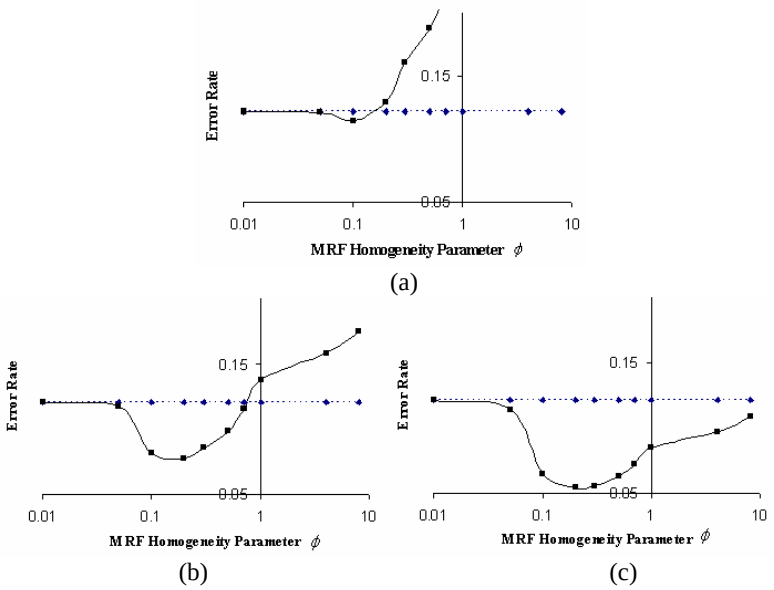


Figure 2: Output Error Rates obtained from technique over 10 iterations varying ϕ (Solid Curve) compared to output from mixture modelling alone (Dotted Curve). Data created with Poisson distributed values of 4 and 10 photons/pixel/integration time (a). FWHM = 0.47 (b) FWHM = 2.4 (c) FWHM = 5.9

Quantitative results were found by comparing returned estimates X with the true scene using the error rate $err = \sum_{i \in S} (1 - M) \Big/ S_n$,where S_n is the total number of pixels and M = 1 if $X_i = \hat{X}_i$, M = 0 otherwise. Minimisation of the error rate corresponds to the best input prediction, i.e. an ideal segmentation should give an error rate of err = 0.

It can be seen in each case that an optimal value for the error rate improves upon the error rate returned from mixture modelling alone. An optimal value of ϕ is shown in each case where the error rate is minimised, i.e. where the Potts model with a value of ϕ best represents the spatial dependencies present in the input image. The figures also show that as the spatial dependencies increase from 2(a) through 2(c), improvements in the corresponding results can be clearly seen. We can see that the optimal error rates returned are improved as the spatial dependencies within the original true scene are increased, and that the error rate is better than that returned with mixture modelling alone over a larger percentage of chosen ϕ. These results are intuitive with what we

would expect, and were replicated for all synthetic imagery with average light levels less than 100 photons/pixel/integration time.

Fig. 3 shows the average percentage improvement of our Poisson MAP-MRF technique over the more widely used Gaussian MAP-MRF technique in terms of optimal error rate returned. Results were obtained from averaging optimal error rates returned from both approaches with 15 different synthetic images. The figure shows that the Poisson approach outperforms the Gaussian approach as the average light level decreases. This follows intuitively, from our earlier investigation of low-light level statistics.

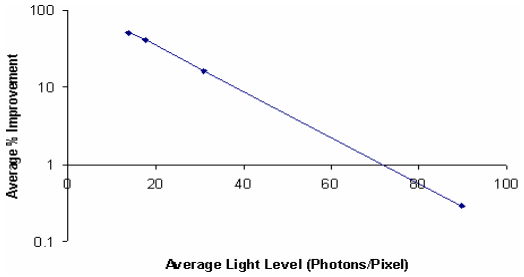


Figure 3: Average % Improvement with Poisson MAP-MRF over Gaussian MAP-MRF as a function of average light level for low-light level Synthetic Imagery

3.3 Segmentation of L3CCD Imagery

Synthetic imagery was then created to approximate the output from a L3CCD sensor with varying mean gains. This was achieved by replicating the avalanche multiplication process applied by the multiplication register of a L3CCD on the low light synthetic imagery.

Fig. 4 shows the average percentage improvement of the Poisson MAP-MRF technique over the Gaussian MAP-MRF technique in terms of optimal error rate returned; with differing mean gains applied.

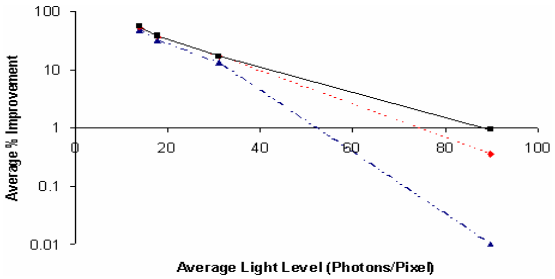
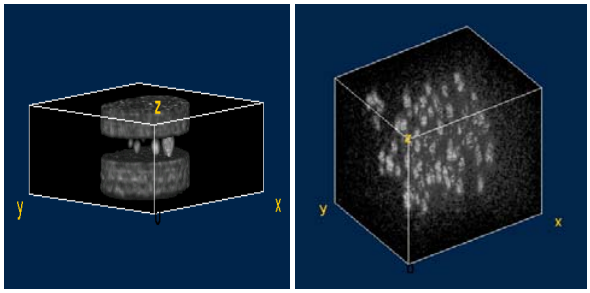


Figure 4: Average % Improvement with Poisson MAP-MRF over Gaussian MAP-MRF as a function of average light level for Synthetic L3CCD Imagery created with varying mean gains of 100 (Solid Curve), 500 (Dotted Curve), and 700 (Dash-Dotted Curve).

The figure again clearly shows that the Poisson approach outperforms the Gaussian approach for all mean gains shown, improving as the average light level decreases. We can also see that, in general, as the mean gain increases, the percentage improvement decreases somewhat due to the fact the distributions present in the data tend to Gaussian as the gain increases.

Finally, the technique has been applied to the segmentation of data obtained from a L3CCD sensor. The 3D dataset of dimension 208*235*100 used for these results consists of Green Fluorescent Protein labelled telomeres in a cell nucleus; acquired via a L3CCD sensor. These are a component of cells which act as a buffer at the end of chromosomes to prevent the loss of genetic information needed to sustain its activities; with implications in anti-aging and anti-cancer therapy [10]. The technique was also applied to a Poisson distributed PET imaging source, of dimension 128*128*35 for comparison; as the technique is applicable to any Poisson distributed dataset. The phantom data used was an oval shaped object containing six fillable spheres of variable diameters filled with fluorodeoxyglucose. Both datasets are shown in Fig. 5.



(a) (b)

Figure 5: (a) 3D PET Dataset (b) 3D L3CCD Dataset

Table 1 shows the error rate obtained from various segmentation techniques applied to both datasets, as compared to manual segmentation.

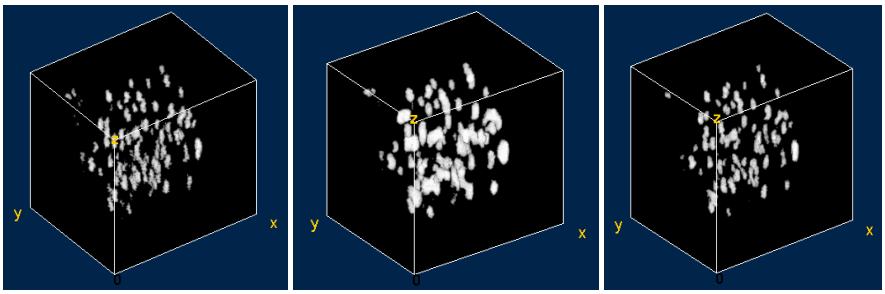
	Otsu (Automated Thresholding) *[*10e-3]*	*Gaussian MAP-MRF* *[*10e-3]*	*Sq-Root Gaussian MAP-MRF* *[*10e-3]*	*Gaussian MAP-MRF with Median Filter* *[*10e-3]*	*Poisson MAP-MRF* *[*10e-3]*
PET Imagery	199	0.653	178	0.687	0.357
L3CCD Imagery	849	1.30	1.10	1.20	0.614

Table 1: Error Rates obtained from various Segmentation Techniques as compared to Manual Segmentation

In each automated MAP-MRF case, the number of total components in an image was chosen via Bayesian Information Criterion (BIC) analysis (3 in each case for the L3CCD data), and the value of ϕ chosen via PLIC analysis. Like PLIC, BIC is an

automated Bayesian technique that considers the ratio of likelihoods of output models to determine the optimal model. Included are the Otsu automated thresholding technique [11], the Gaussian MAP-MRF technique, the Gaussian MAP-MRF technique applied to the square-root values of the original dataset, and with a median filter applied; and the Poisson MAP-MRF technique.

The results clearly show that the Poisson MAP-MRF technique provides a result closer to that obtained via the manual segmentation technique than the Otsu and Gaussian MAP-MRF techniques in both cases. The Poisson MAP-MRF technique also results in less noisy isolated pixels, and clearer isolated telomere bodies than the corresponding Gaussian results for the L3CCD data, as shown in the qualitative results of Fig. 6. The reason for this is that assuming Poisson statistics allows us to better model the true distributions present in the L3CCD datasets.



(a) (b) (c)

Figure 6: Segmentation of L3CCD Dataset (a) Output From Manual Technique (b) Output from Gaussian MAP-MRF Technique (c) Output from Poisson MAP-MRF Technique

Both fully-automated techniques gave comparable computational times of approximately 60 secs per ICM iteration for a 14-bit image of dimension 208x235; using a 2800Mhz x86 GenuineIntel processor. To conclude, better qualitative segmentation was found with the Poisson technique over the Gaussian using a fully automatic technique, with comparable processing times.

4 Conclusions and Future Work

We have presented a novel technique, using a Poisson MAP-MRF approach for the segmentation of low-light imagery which, to our knowledge, has not previously been applied to the described classes of imagery. We have shown that the technique provides improvement over a simple mixture modelling approach, and that improved results are seen as the spatial dependencies within the data are increased. The technique shows improvement over a Gaussian MAP-MRF approach at low light levels, and has been applied to real L3CCD data, with successful results. Future work shall include further investigations to take fuller advantage of the distributions present within L3CCD datasets. Application of the technique to 4D imagery shall be implemented.

Acknowledgements

Funding was provided by Andor Technology and DEL NI. PET datasets were provided by the Northern Ireland Cancer Centre.

References

[1] D. Montgomery, A. Amira, & F. Murtagh, "An automated volumetric segmentation system combining multi-scale and statistical reasoning," *Proceedings of the IEEE International Symposium on Circuits and Systems, Kobe, Japan*, 4, pp. 3789-3792, May 2005.

[2] Z. Peng, W. Wee & J. Lee, "MR brain imaging segmentation based on spatial Gaussian Mixture Model and Markov Random Field," *IEEE International Conference on Image Processing*, 2005.

[3] H. Deng & D.A. Clausi, "Unsupervised Image Segmentation Using A Simple MRF Model with A New Implementation Scheme", In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, 2, pp. 691-694, IEEE, August 2004.

[4] J. Goldberger & H. Greenspan, "Context-based segmentation of image sequences," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(3), March 2006.

[5] G. Basden, C. A. Haniff & C. D. Mackay, "Photon counting strategies with low-light level CCDs," *Mon. Not. R. Astron. Soc*, 345, pp. 985-991, 2003.

[6] A. Dempster, N. Laird, & D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, 39(1), pp. 1-38, 1977.

[7] D. Stanford & A. E. Raferty, "Approximate Bayes factors for image segmentation: The pseudo-likelihood information criterion (PLIC)," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(11), November 2002.

[8] S. Z. Li, *Markov Random Field Modelling in Computer Vision* New York: Springer-Verlag, 2001.

[9] J. Besag, "On the statistical analysis of dirty pictures," *Journal of the Royal Statistical Society, Series B (Methodological)*, 48(3), pp. 259-302, 1996.

[10] Y. Cong, W. E. Wright, & J. W. Shay, "Human telomerase and its regulation," *Microbiology Molecular Biology Review*, 66(3), pp. 407-425, September 2002.

[11] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1), pp. 62-66, 1979.

Real-Time Humans Detection in Urban Scenes

Julien Bégard, Nicolas Allezard and Patrick Sayd
CEA, LIST,
Boîte Courrier 65, Gif sur Yvette, F-91191 France
(julien.begard,nicolas.allezard,patrick.sayd)@cea.fr

Abstract

We address the issue of real-time pedestrians detection in a urban environment. This is a challenging task owing to the high variability of appearances and poses that humans can have and to the complexity of backgrounds. We propose a solution made of gradient-based local descriptors combined to form strong classifiers and organized in a cascaded detector. We developed for this an extension of the Histograms of Oriented Gradients (HOGs) and added a new component to the histogram which represents the *strength* of edges or the *amount of information* in the histogram support. We also implemented a learning algorithm based on Real Adaboost where two phases – selection first, then refinement of weights – provide more robustness to the detector. We evaluated our system by comparing it to the cascaded detector of Haar features of Viola & Jones [7] and to the SVM of HOGs features of Dalal & Triggs [1]. To ensure an equitable and valid comparison, we used the database proposed in [1]. Our system outperforms them in detection results and in time needs.

1 Introduction

The human class in object detection is probably one of the more difficult because of the variability of appearances and poses that humans can have. The descriptor used to describe and to detect human silhouettes has to capture finely the good characteristics to assure an efficient detection. Moreover, applied to an urban context, the detection chain has to be as robust as possible and has to run in real-time. We developed a human detection system based on Histograms of Oriented Gradients (HOGs) features learnt by a cascaded boosting procedure which performs better than Viola-Jones [7] Haar+Adaboost cascade (**system 1**) and Dalal-Triggs [1] HOG+SVM (**system 2**). We adopted the cascade approach from [7] with some modifications which tend to make the system more robust and we worked with HOGs-like descriptors including gradient strength information. This paper is organized as follow: we first make an overview of previous work on human detection in 2. Then we present briefly our system in 3, experimental results are exposed in 4 with a comparison with other state-of-the-art methods. In 5, we present the performance of our system through several parameters study and we finally conclude and discuss about future work in 6.

2 Previous Work

Lot of work has already been done in Computer Vision based detection systems and an extensive literature accompanies this work. Papageorgiou *et al* [6] presented a detector which infers a pedestrian model from positive and negative examples by the mean of a polynomial SVM and uses *quadruple density* Haar Wavelets as a pattern descriptor. An improved version has been developed by Deporrtere *et al* [2], principally with dimension reduction thanks to Adaboost features selection and SVRM[1] model generation. Viola *et al* [7] introduced a Haar-like wavelets coarse-to-fine cascaded face detector combined with an efficient features extraction method – the integral image – and extended this system to take into account motion cues in video scenes [8]. The system developed by Gravila *et al* [3] computes the chamfer distance to perform a shape-based pedestrian detection and validates the detection with textures classification from a neural network and stereo verifications. Mikolajczyk *et al* [5] assembled seven part detectors (frontal and profile head, face and upper body, legs) with Bayesian decision rules. Dalal and Triggs [1] proposed a detector build on a SVM learning machine and Histograms of Oriented Gradients (HOG) through a very interesting study of their implementation issues. A more challenging goal is aimed by Leibe *et al* [4] and Wu and Nevatia [9] who tried to detect partially occluded pedestrians in crowded scenes. The formers considered the aggregation of local and global cues while the latter used *edgelet* features learnt by a boosting method.

3 Overview of the System

In this section, we describe briefly our system schematized in figure 1. More details for each parameters are given in section 5.

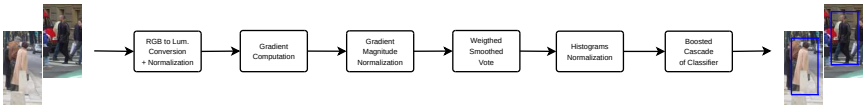


Figure 1: Detection chain of our system.

We developed a system that can achieve real-time pedestrians detection in urban scenes with very low false-positive rates. Real-time detection is an important point for this system which could be adapted to a future embedded version.
Our method is based on a true/false classification obtained by an evaluation of local descriptors computed densely on input images through a boosted cascade of classifiers.

3.1 Local Descriptors

Distinction and recognition of objects in static scenes can be obtained from shape contours. HOGs work well to capture such information from an image but it lacks of a clue, which could be named the *contour strength*, to make difference between acute and loose edges. To correct this drawback, we included the gradient magnitude as a new component in the histogram so that our histograms are now made of 9 magnitude-normalized orientations bins – each one 20°wide – plus 1 bin for the gradient magnitude. The orientations

[1] Support Vector Regression Machines

are unsigned so that we do not make difference between a dark-bright transition and a bright-dark one (this assumption makes sense with the high color and texture variability of human appearances: hair, skin, clothes, etc.). To reduce aliasing, we use a kind of simple smoothing to compute the histogram components by giving a fraction x of the vote to the corresponding bin and a fraction $1-x$ to the nearest bin where $x \in [x_{min}, 1]$. x_{min} depends on the angle threshold θ_T above which we consider a vote belonging to one and only one bin ($x = 1$). In fine, we obtain heterogeneous descriptors, computed in a dense way, that can code the orientations and the *strength* of a shape.
Input images are represented in grayscale color space and they are first normalized along the luminance in areas where there are enough gradient information. This preprocessing provides to the system some robustness to illumination conditions. Gradients are computed with an optimized implementation of the Deriche operator and components votes are efficiently accumulated thanks to the integral image techniques.

3.2 Learning Procedure

Pedestrians characteristics are extracted and learnt by a cascaded boosting algorithm like the one used in [7]. In our case the learning algorithm has to face with a very high dimensional space made of almost ten thousand of components for each positive and negative example. The Adaboost algorithm is used to select the relevant components and to train classifiers from them like **system 1**. The boosting algorithm builds, for each stage, a strong classifier from a weighted selection of features – the weak classifiers – and stop selecting features when the training errors drop below a threshold.
We changed a bit this procedure to robustify the resultant strong classifier: we first select n weak classifiers, each of them is a component of an histogram. Viola's algorithm would stop there but ours comprises another loop of m rounds of boosting to compute more finely the weights of the n selected components. The difficulty is to find the two optimal parameters n and m to prevent the system from over-learning, which could lead to a decrease in the performances.

4 Results and Comparison with Different Systems

4.1 Databases

Our system is dedicated to human detection in urban scenes. To train and test it, we acquired several hours of video with a video-camera fixed on a car driven in cities (see fig. 2 for examples of image). We obtained very good results on this database. Although we could use these datas in this paper, we would prefer using the database available in [1], where the background is not especially urban, to make more neutral comparisons between the different systems. This database contains upright humans with various poses, clothes, backgrounds and light conditions. Some samples have partial occlusions. There are 2478 96×160 positives examples for training (1239 + left-right reflexions) and 1126 70×134 positives examples (563 + reflexions) for testing. People in these images are 64×128 sized. There is also a free-person set of 1218 images for generating the negative examples.

4.2 Methodology.

We trained and tested our detector on this database in two different ways. The system presented in [7] is based on a cascade of boosted classifiers whereas the system in [1]

Figure 2: Samples of images in our urban database.

Figure 3: Some samples from the database we used.

is based on a linear SVM. Although the cascade hierarchy provides a faster behavior to the system, it also decreases a little the detection rate. Indeed, several stages have more chance to reject wrongly good candidates than only one stage. This is why we separated our comparisons in two distinct parts: the first comparison will be between the cascades of classifiers, a Haar-based from Viola-Jones and our cascade. The second comparison will compare the SVM+HOGs system from Dalal-Triggs with a *one stage* cascade trained with our method. For each case, we give ROC curves with Miss Rate ($1-$Recall) versus False Positive Per Windows (FPPW) rate.

4.3 Results & Comparisons.

Boosted cascade of Haar classifiers vs. improved boosted cascade of HOGs. We used Intel OpenCV implementation of Viola-Jones detector and we trained it with the datas we described above. We gave the same parameters to the two training programs: 2478 positives samples and 10000 negatives samples and we built two cascades made of 10 stages with the Real version of Adaboost. Minimum hit rate and maximum false alarm rate are respectively fixed to 99.5% and 50%. Training has been done on an Intel Xeon Processor at 3.0GHz equiped with 4Gb of memory and it took several days (almost 1 week) for **system 1** whereas ours needed only a few hours (almost 2 hours). The final cascades used 397 different weak classifiers for **system 1** and 169 for ours.
Now let's talk about results and performance. Figure 4 clearly shows that our cascaded pedestrian detector is better than **system 1**. Further studies shows that **system 1** has difficulties to detect correctly pedestrians present in the database we used. This is due to the variety of backgrounds and parasitic elements like traffic signs, billboards, etc. By the same way, those same elements add false alarms leading to an overall decrease of the performance. Considering processing times, it took 46.4 ms for **system 1** and 27 ms for our system to scan a 320×240 image with the same parameters (scale factor of 1.2, step size of 1.5). The figure 4 shows the ROC curves for **system 1** and our system.

HOGs+linear SVM vs. *1-stage* HOGs+Boosting. We used the latest system avail-

able described in [1] to compare it with our method. We let the default parameters as there are supposed to be the best for this system and we trained it with the same examples set, which took several hours (almost 3). To have a comparable system, we trained a *one stage cascade* with our own program with 50 weak classifiers. This was done in less than an hour.
Regarding the results, our system quickly outperforms the system based on HOGs+linear SVM and provides a near perfect detection with an acceptable false positive rate. Our system is a little bit less performant for FPPW under $5E-5$ but above, it is better than **system 2** and reaches perfect detection for false alarm rates after $6E-4$. See fig. 4 for the whole ROC curves. We also compared processing time on 320×240 images with a dense scan of almost 5500 detection windows: **system 2** took 3 seconds whereas ours needed only 229ms. Remind that the cascaded detector needed only 27ms: this shows how the cascade structure is efficient to reduce detection times.

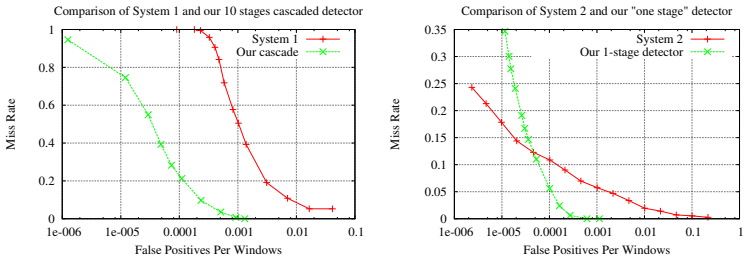


Figure 4: Performances of **system 1**and our cascade (left) and performances of **system 2**and our *1-stage* detector (right). In both cases, our method outperforms the other systems.

Figure 11 shows examples of output of our systems on some test images of the database from [1].

5 Performance and Parameters Study

In this section, we study the impact of some parameters on the system. After addressing the issue of gradient computation method, we justify the use of the gradient *strength* as a new component in the histogram and then study the drawing of the dense grid on which histograms are computed. We expose then the initialization of the learning algorithm and the manner in which a strong classifier is trained. After that, we discuss of the existence of an optimal number of stage in a cascaded detector and we finally study the structure and characteristics of the resultant detector.

5.1 Gradient Computation

The detection chain almost begins with gradient computation so that the detector results are closely dependent with the way in which they are computed. We tested several gradient computation methods to select the best result/time ratio as we want our system to be real-time. Rapid gradient computations could be done with simple small mask derivation (1D: $[-1,0,1]$, 2D: Sobel) and also with the Deriche recursive operators.

The only parameter for Deriche's operators is α which drives the *smoothing* phase of the filter. The smaller α is, the stronger the smoothing will be. But, as our descriptor finds complementary information on the gradient strength, α must not be too small.
We looked then at the influence of the size of the normalization windows. As expected, the evolution of the results reached an optimum as the size increases. We started from 10 pixels to 30 pixels and the optimum is between 15 and 20 pixels. Note that this window is as broad as an arm/leg of human on the training samples.

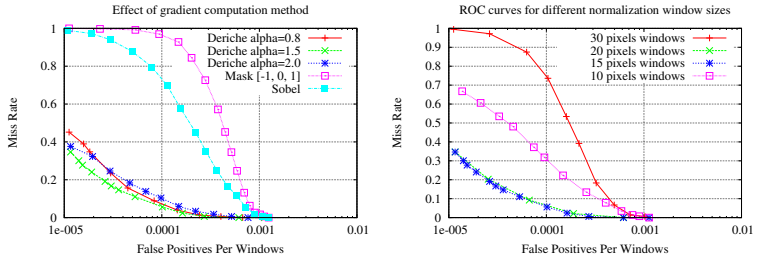


Figure 5: Effect of gradient computation method (left) and of normalization window size (right).

5.2 Contribution of the Gradient Magnitude in the Descriptor

We now study the contribution of the gradient magnitude that we called the *contour strength*. This is done by comparing two detector: one with classic HOG descriptors and the other with HOG+magnitude. The two systems differ only on the descriptors used, they are trained with the same parameters and on the same database. Figure 6 clearly shows that including the magnitude of the gradient as a new component of the histogram improves the results.

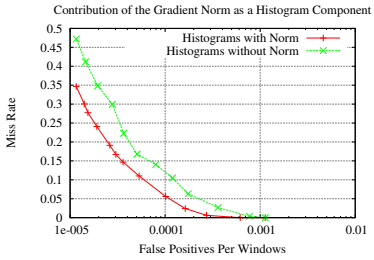


Figure 6: Comparison between HOGs with norm and without norm. There is a positive gain of almost 0.1 between the two descriptors at no additional calculation cost.

5.3 Drawing a Dense Grid on the Image

Our system is sensitive to the gradients computation method, but obviously it is also sensitive to the manner used to *draw* a dense grid on inputs image. We evaluated different

schemes with fixed sizes of windows (8×8, 16×16, 24×24) and with relative sizes of windows ($[0.1 \rightarrow 0.9] \times$ *ImgSize* with position steps of 0.1 and a scale factor steps of 0.2 for example). Our tests arrive at the conclusion that a multiscale grid is better and that it is preferable to let the learning algorithm run in a very high dimensional space than reducing this dimension by fixing arbitrarily restrictive sizes for the computation grid. Figure 7 shows this results.

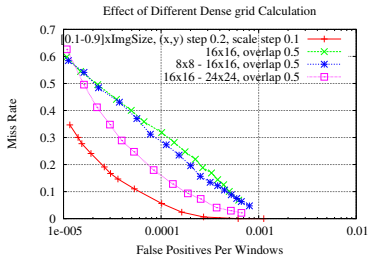


Figure 7: The results of the detector depends on how the dense grid of histograms is computed. A multiscale grid is better and the results are even more better if the histograms are computed on regions of the size of a human limb.

5.4 The Cascaded Detector

We noticed several important points as we worked on our cascaded pedestrian detector. First is the issue of the initialization of the learning algorithm. There are two possibilities to initialize the weights of the training samples: 1. $w_+ = w_- = \frac{1}{N_+ + N_-}$ or 2. $w_+ = \frac{1}{N_+}$ *and* $w_- = \frac{1}{N_-}$ with $N_{+/-}$ the number of positive/negative examples. The second initialization method leads in fact to two cases depending on the amount of positive and negative examples. We could intuitively think that giving a better weights to positive (or negative) examples will influence the system so that it focuses its discriminative work on them. This is true but finally, this does not improve the results at all. It is better to let the system find his own rules with an equitable initialization.
The second point concerns the number of stages of the cascade. As we know, the cascade structure is useful to accelerate the processing as it uses a coarse-to-fine approach. Although this technique degrades a little the detection results, the gain in performance is satisfactory enough to justify this loss. Nevertheless, the cascade length has a limit beyond which the system speed stagnates whereas the results decrease too much. Thus, this indicates that there would be an optimal number of stages for building a cascaded detector. We verified this and tried to show the independence of this number of stages to the testing set. Those two points are illustrated by figure 8.

5.5 Training a Strong Classifier

The learning algorithm we developed for our system is quite similar as the one used in **system 1**. But, whereas **system 1** lets the Adaboost algorithm run until it found the required features, we force our algorithm – a classical Real Adaboost – to begin another loop of rounds of boosting after the selection phase. The idea is to first select the best

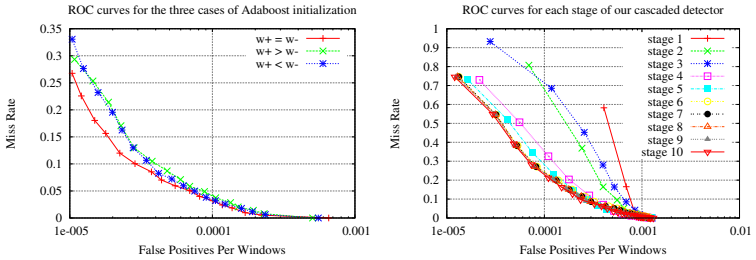


Figure 8: Left: effect of the initialization of the learning algorithm. Right: The ROC curves for each stage of our cascaded detector. The results quickly reach an asymptote.

features and then to refine the weights as precisely as possible.
For this, we have to determine previously the number of descriptors and also the number of additional rounds of boosting to calculate the weights. So we study the influence of those two parameters on our system whose results reach an asymptote as they increase. We noticed that a strong classifier built with very few descriptors is very fast, of course, but useless because it is not discriminant enough. On the other hand, a big strong classifier has good results but is also very slow. As an example, we built our *one stage* detector with 50 descriptors boosted by 10 rounds/descriptors.

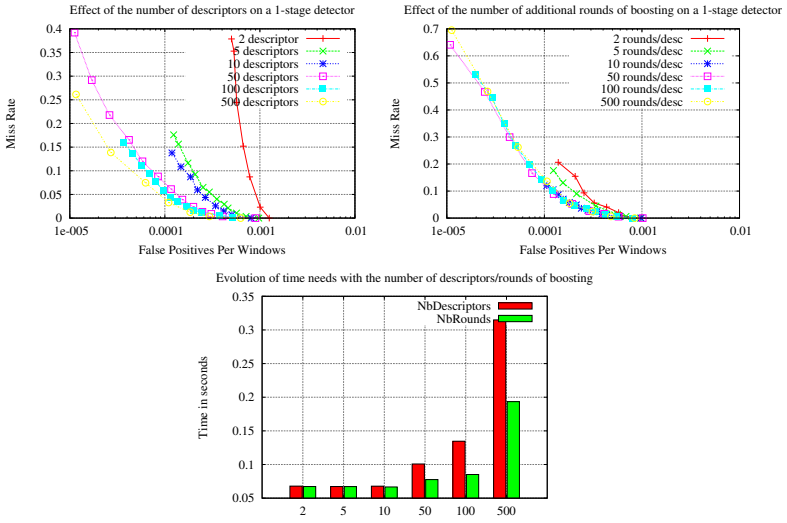


Figure 9: Effect of the number of descriptors and the number of additional rounds of boosting on a 1-stage detector. We choose, as a compromise between detection rates and time needs, 50 descriptors for 10 rounds per desc. of additional boosting iterations.

5.6 Post Learn Study

Results and detection speed are good indicators to evaluate our system, but we would like to go further and to see how our learning machine built the detector. We observed that *stronger* histograms are chosen more often than others. In training samples, this corresponds to select the edges of the silhouettes we want to learn (see fig. 10). Moreover, when we focus on descriptors that are the most chosen, or on descriptors that are the most discriminant – they have the strongest positive/negative vote – we noticed that they are localized in regions like *head*, *shoulders*, *legs* and the *crotch*. That is to say characteristic regions that allow to recognize a pedestrian in an image. Finally, histograms from regions as broad as a human limb are also selected more often than others.

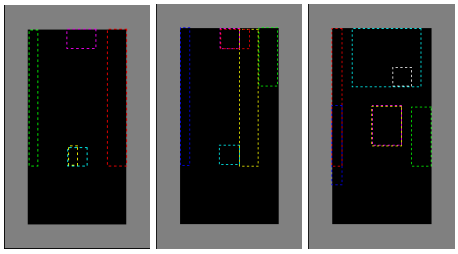


Figure 10: Examples of support of HOG selections. The main regions are near the head, shoulders, legs and also the crotch. Pedestrians are in 64×128 bounding boxes (black) inside 96×160 images (gray).

6 Conclusion and Future Work

We have proposed a system that detects pedestrians – vulnerable users of the road – in real time. To build this system, we used HOGs local descriptors, computed on a dense overlapping grid and learnt by a boosting algorithm whose output is a cascaded detector. We have shown that considering the gradient magnitude as a new component in the HOGs provides new information to the learning algorithm and improves dramatically the results. We have also presented a learning procedure based on Adaboost made of two phases – a feature selection phase followed by the precise refinement of the weights – which adds robustness to the system while improving the detection rates. We evaluated our system and compared it to Viola-Jones one and Dalal-Triggs one in equitable conditions and showed that our detector performs the best.

Although detection rates and speeds are satisfactory, there is obviously still room for optimizations. The computation of the score of the strong classifiers is far from being efficient since we did not limit the displacements in the image. In the same way, the boosting algorithm we used is an accumulating algorithm. That is to say the only action it does is to add features to the classifier and adjust the weights. We could employ another algorithm that, for every iteration, decides whether to add a new feature or remove an old feature. Context in the image also provides a lot of information and could be used, for example, to remove false alarms or to get rid of some partial occlusions. Finally, we currently begin to study how to combine the histograms together to reinforce the selection of the learning algorithm. The idea beyond this is intuitive: an histogram is selected

because it contains discriminative information. So its direct neighbors could also contain this information, even in a lower quantity. We call this *Adjacent HOGs*.

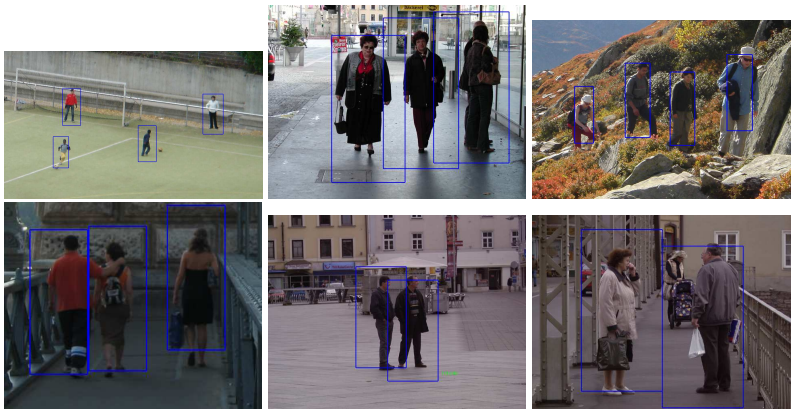


Figure 11: Some examples of detection on full size images.

References

[1] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume II, pages 886–893, 2005.

[2] V. Deporrtere, J. Cant, B. Van den Bosh, J. De Prins, R. Fransens, and L. Van Gool. Efficient pedestrian detection: a test case for svm based categorization. In *Workshop on Cognitive Vision*, 2002.

[3] D.M. Gravila, Giebel J., and Munder S. Vison-based pedestrian detection: The protector system. In *IEEE Intelligent Vehicles Symposium*, 2004.

[4] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *CVPR*, volume I, pages 878–885, 2005.

[5] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In *ECCV*, volume I, pages 69–81, 2004.

[6] C. Papageorgiou, T. Evgeniou, and T. Poggio. A trainable pedestrian detection system. In *IEEE Intelligent Vehicles Symposium*, pages 241–246, 1998.

[7] P. Viola and M. Jones. Robust real-time object detection. In *International Journal of Computer Vision*, 2002.

[8] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *ICCV*, 2003.

[9] B. Wu and R. Neviata. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *ICCV*, pages 90–97, 2005.

Class-Specific Binary Correlograms for Object Recognition

Jaume Amores[1], Nicu Sebe[2], Petia Radeva[3]
[1] IMEDIA Research Group, INRIA, France
[2] Univ. of Amsterdam, The Netherlands
[3] Computer Vision Center, UAB, Spain

Abstract

This paper presents an efficient object-class recognition approach based on a new type of image descriptor: the Class-Specific Binary Correlogram (CSBC). In our representation, the image is described by a collection of CSBCs, where each one encodes the spatial distribution of class-specific features around a particular reference point. This representation is obtained by first performing an automatic selection of class-specific features from a vocabulary, and then extracting collections of binary correlograms that encode, at the same time, detected object parts and their spatial distribution around multiple points of the image. Our descriptors live in high-dimensional spaces (in the order of 10K dimensions), but they are very sparse. We show that efficient learning and matching procedures can be obtained for such a representation if we use, first, fast feature selection techniques specific for binary features, and then Boosting integrated with an appropriate Inverted File data organization. The proposed strategy works with weak supervision, outperforms state-of-the-art bag-of-feature methods, and it is more accurate and computationally more efficient than well-known geometrical-based methods, including our previous work on Generalized Correlograms (GCs) [1].

1 Introduction

In this paper we deal with the problem of Object-Class Recognition, which has received much attention recently [6, 14, 16, 15, 12]. The objective is to detect the presence or absence of objects from a desired category (for example 'car') in the images presented to the system. The problem is challenging due to the large variability of object appearance across instances of the category, added to the variability of pose and illumination, partial occlusion and clutter.

A fundamental issue is how relevant information is extracted from the image and how it is efficiently managed in order to first learn a model of the desired category, and then scan for instances of this model in new images. There is broad consensus upon the suitability of describing the local features and their spatial relation: local features are more robust against clutter than global signatures, and considering the mutual position of features significantly increases the distinctiveness of the representation. However, using spatial relations usually leads to much higher costs in the learning and matching procedure. The common approach is to use graph-base representations where local features and

spatial relations are described respectively by nodes in the graph and arcs between nodes. Despite its flexibility, the problem with such an approach is that learning and matching graph representations is known to be very expensive, even if we use fast optimization procedures [7, 4, 9].

Instead of representing these two types of information (local information about features and geometrical information about the mutual position of these features) by two separate entities (nodes and arcs), we proposed in [1] a Generalized Correlogram (GC) that encodes in the same feature vector the local information describing what are the local features in the image and, at the same time, the geometrical information describing the mutual position of these features. The advantage of such a representation is that, by simply comparing two feature vectors we take into account simultaneously the similarity of local features and their spatial distribution. This allows us to employ fast matching techniques that quickly consider the relevant information.

The Generalized Correlogram from our previous work [1] is a joint distribution of local and geometrical attributes, similar in spirit to previously defined correlograms in the literature [10, 3], but specially suitable for the object-class recognition task. It was defined to be generic in order to be used in retrieval of large databases, where the descriptors of database images were extracted off-line, i.e. regardless of the query the user was interested on. In this paper, we explore an alternative framework and introduce a Class-Specific Binary Correlogram (CSBC), which varies depending on the object category to be recognized. Along with it, we describe fast learning and matching procedures that permit to efficiently manage this type of description by exploiting its binary nature and its high degree of sparseness.

The rest of the paper is organized as follows: section 2 describes the image representation, section 3 describes the construction of model contexts in the learning the object class, section 4 describes fast procedures for learning our representation, section 5 describes the matching step with Inverted Files, section 6 shows results and section 7 concludes.

2 Image representation

The image is represented by a collection of Class-Specific Binary Correlograms (CSBC), each one describing the spatial distribution of class-specific features around a certain reference point of the image. For this purpose, the method: i) extracts a set of local features which convey relevant information about the object category, ii) detects the presence of these class-specific features in each image; and iii) extracts a set of correlograms, each one describing the mutual position of detected features relative to a certain reference point of the image. As we will see, our CSBC descriptors make use of a log-polar spatial quantization [3] that make them semi-local, so that CSBC focus on the local information around a certain reference, and incorporates the contextual properties (how the rest of detected object parts are distributed around the reference) in a smoothly decreasing degree of attention. Let us describe each step of the representation procedure.

2.1 Extraction of class-specific local features

Local features specific of the object category are gathered by using feature selection over a large pool of features extracted from training images. The extraction procedure is divided in the following stages: i) extraction of a large pool of local features, referred to as

"dictionary" in the bag-of-features paradigm [13, 21, 5] and ii) selection of a reduced set of class-specific features that convey relevant information about the object class.

2.1.1 Building a dictionary of local features

Let the training set T consist of N images $T = \{I_1, \ldots, I_N\}$, and let the i-th image $I_i \in T$ have an associated set of local descriptors (features) $L_i = \{\vec{l}_{i1}, \ldots, \vec{l}_{iK}\}$ extracted from I_i. These local descriptors can be extracted either from interest points detected in the image or by simply performing a random sampling in both position and scale. In our implementation we employ the interest point extractor of Kadir et al. [11], which obtains a set of points with maximum entropy in scale-space, and then take the K most salient points. Finally, the vocabulary V is built by gathering all the local descriptors extracted from each training image: $V = \cup_{i=1}^{N} L_i$.

2.1.2 Selection of class-specific features

Given a large pool of features V, we want to select a reduced set which conveys the maximum amount of information about our particular object category. Different mechanisms have been employed in the literature for this purpose. The standard one finds class-specific features by observing the frequency of occurrence of each feature from the vocabulary in each image from the training set. Intuitively, those features that are more frequently found in foreground images than in background ones will convey information about our object class. This approach is followed in the well-known bag-of-features paradigm [13, 21, 5], where each image I is described by a histogram h that counts the number of times that each feature from the vocabulary V is matched with some feature of I. A problem observed this approach is the sensitivity to the number of bins utilized in the histograms.

In this work we follow an alternative method. For each image I_i of the training set, we compute a vector of distances $\vec{d}_i = (d_1, d_2, \ldots, d_M)$ where the j-th element measures the distance $d(\vec{v}_j, I_i)$ between the j-th feature $\vec{v}_j$ of the vocabulary and the image I_i. This distance is defined as the minimum distance from $\vec{v}_j$ to all the features found in I_i: $d(\vec{v}_j, I_i) = \min_{\vec{l} \in I_i} \|\vec{v}_j - \vec{l}\|$. In other words, for each $\vec{v}_j$, we match it against the feature $\vec{l} \in I_i$ with minimum distance, and keep the distance score. Intuitively, if $\vec{v}_j$ is some characteristic feature of the object, most of the foreground images in the training set will have some local feature that is similar to $\vec{v}_j$, whereas background images will have features that are dissimilar. Therefore, distances to foregrounds $d(\vec{v}_j, F)$ will be generally smaller than distances to backgrounds $d(\vec{v}_j, B)$.

Once we represent each training image I_i by a distance vector $\vec{d}_i$, we can employ several techniques to select class-specific features. If we use maximum information, we will obtain a method very similar to the method used by employed by Ullman et al. [18] for selecting their class-specific image fragments. Alternatively, we can use Boosting with decision stumps in order to obtain a small set of informative features that maximally complement each other in the classification task. The latter has the advantage of being computationally much more efficient, and it is the method that we use in this work. This method was followed by Opelt et al. in [14], although they did not use it for feature selection, but directly for obtaining a classifier.

Selecting class-specific features with AdaBoost and decision stumps as weak classifiers is straightforward, once we have each training image I_i described by the distance

vector $\vec{d_i}$. We only need to call the standard AdaBoost procedure with the training pairs $(\vec{d_i}, y_i), i = 1, \ldots, N$ where y_i is the label of the image I_i. We used the AdaBoost with decision stumps algorithm described by Viola et al. in [19], we do not reproduce it here due to lack of space.

As a result, AdaBoost obtains an ensemble of decision stumps $f_1, \ldots, f_K$, where K is a parameter introduced to AdaBoost. The k-th decision stump $f_k(\vec{d})$ is the function:

$$f_k(\vec{d}) = \begin{cases} 1 & if \quad d_{i_k} < \theta_k \\ 0 & otherwise \end{cases},$$

where i_k and θ_k are parameters defining f_k. In our case, $\vec{d}$ is the vector of distances computed for some image I, and the element i_k of this vector, d_{i_k}, represents the distance of image I to the feature $\vec{v}_{i_k}$ from the vocabulary:$d_{i_k} = d(\vec{v}_{i_k}, I) = \min_{\vec{l} \in I} \|\vec{v}_k - \vec{l}\|$. Therefore, in our case the following definition is equivalent:

$$f_k(I) = \begin{cases} 1 & if \quad \exists \vec{l} \in I : \|\vec{l} - \vec{v}_{i_k}\| < \theta_k \\ 0 & otherwise \end{cases}$$

In other words, the function f_k is activated over the image I iff there is an image feature $\vec{l} \in I$ whose distance to the vocabulary feature $\vec{v}_{i_k}$ is lower than the threshold θ_k. In our setting, Boosting selects those features $\vec{v}_{i_k}$ that better separate positive and negative images respect to their distance to $\vec{v}_{i_k}$. In other words, Boosting selects those features $\vec{v}_{i_k}$ that match (i.e. have a small distance) with some feature in every foreground image and that, at the same time, do not match (i.e. have a big distance) with all the features of background images, which is exactly what we need, and additionally it outputs the matching thresholds θ_k found to be statistically more consistent.

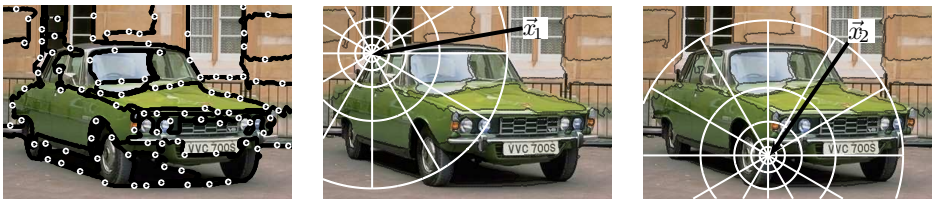
Finally, let us express as the output of this step as the set $C = \{\vec{c}_k\}_{k=1}^K$ of features, so that $\vec{c}_k = \vec{v}_{i_k}, k = 1, \ldots, K$. We will refer to features in $\vec{c} \in C$ as *class features*, because they convey information about our object class. Associated with C, we obtain a set of thresholds $\Theta = \{\theta_k\}_{k=1}^K$, where θ_k defines the matching threshold of the class feature $\vec{c}_k$, i.e. $\vec{c}_k$ matches with all the image features from I with distance to $\vec{c}_k$ lower than θ_k.

2.2 Extraction of Class-Specific Binary Correlograms

In our implementation, given image I we extract a dense set of image local descriptors sampled every 4 pixels in 7 different scales that range from 0.5 to 2 times the size of the image, with linear increments. We use a very basic type of local descriptors reported in [6]: local gray-level windows re-scaled to size 11×11 and projected by PCA to 15 dimensions, parameters used in [6]. Much better results can be obtained with more sophisticated descriptors, but our interest in this work was to measure the relative performance compared to previous methods with the same type of local descriptors.

Given the image I, let the class feature $\vec{c}_k \in C$ match with the set of image features $M_k = \{\vec{m}_j^k\}_{j=1}^{N_k}$ from the image, and let us define $P_k = \{\vec{p}_j^k\}_{j=1}^{N_k} \subset R^2$ to be the spatial positions from which these matching features are extracted, i.e. $\vec{p}_j^k \in P_k$ is the spatial position of matching feature $\vec{m}_j^k \in M_k$.

Let $X = \{\vec{x}_i\}_{i=1}^R$ be a set of reference points sampled from the image I. For each reference point $\vec{x}_i \in X$ we extract a Class-Specific Binary Correlogram h_i as follows. Let the spatial relation $(\vec{x}_i - \vec{p}_j)$ from the reference to any point $\vec{p}_j$ be expressed in polar



(a) (b) (c)

Figure 1: (a) Sampled set of points taken as reference. (b)-(c) Log-polar spatial quantization of our descriptor given two different references $\vec{x}_1$, $\vec{x}_2$.

coordinates $(\alpha_{ij}, r_{ij}) : \alpha_{ij} = (\widehat{\vec{x}_i - \vec{p}_j})$, $r_{ij} = \|\vec{x}_i - \vec{p}_j\|$. The angle and radius are quantized into n_α and n_r bins respectively. Let A_u be the u-th bin of angles, $u = 1, \ldots, n_\alpha$, and let R_v be the v-th bin of radius, $v = 1, \ldots, n_r$. We use the same log-polar spatial quantization as used by Belongie et al. for their Shape Context descriptor in [3], see fig. 1(b)-(c).

The class-specific correlogram h_i associated with $\vec{x}_i$ measures the distribution of points $\vec{p}_j^k$ according to their position relative to the reference $\vec{x}_i$ and the index k of the class feature. The distribution is estimated by the histogram:

$$h_i(u,v,k) = \#\{\vec{p}_j^k \in P_k : (\widehat{\vec{x}_i - \vec{p}_j^k}) \in A_u, \|\vec{x}_i - \vec{p}_j^k\| \in R_v\}, \quad (1)$$

Note that index k provides local information, as it indicates the class-specific local feature that is detected at position $\vec{p}_j^k$. Therefore, h_i measures a joint distribution of geometrical properties (the quantized spatial relations to the reference) and local properties (the type of matching feature). Instead of considering h_i, we binarize it to obtain a binary correlogram. This permits to compact considerably the correlogram, and allow us to employ fast techniques based on binary features. Also, in practice we did not observe a significant difference in performance, although a systematic comparison was not performed. In the end, we obtain a CSBC $h_i(u,v,k)$ that flags those spatial bins (u,v) where class feature $\vec{c}_k$ is detected.

As said before, the angle and radius are quantized using the log-polar quantization of [3] (see Figs. 1(b)-(c)), which makes the descriptor more sensitive to local context. The angle is quantized into $n_\alpha = 12$ bins, and the *logarithm* of the radius is quantized into $n_r = 5$ bins. If we use K class features, the CSBC h_i has $12 \times 5 \times K$ dimensions. For example, for $K = 250$ class features, we obtain 15000 dimensions. However, the descriptor does not occupy much space: first, it can be represented in a bit-wise manner, i.e. if we use integers of 32 bits the CSBC occupies only 15000/32 = 469 integers. Further, most of the elements are 0, so that we can use a sparse format which further reduces the space to a few integers.

Reference points in X are sampled from contour points extracted from the image, and we keep the R points with maximum distance to each other, so as to cover the image from every possible angle, a more detailed explanation can be found in [1], for an illustration see fig. 1(a).

In the final image representation we use a multi-scale representation, and express the image description as:

$$\mathscr{H} = \{H_s\}_{s=1}^{S}$$
$$H_s = \{\vec{h}_{is}\}_{i=1}^{R}$$

where H_s is a set of CSBCs scaled according to the s-th scale, and S is the total number of scales. CSBCs are scaled by normalizing the radius of the difference vectors expressed in polar coordinates.

3 Building a collection of model contexts

As we saw, images are represented by collections of correlograms, where each one represents a particular *image context*, i.e., how different types of local features are spatially distributed around a particular reference point. Given this representation, we also express the learned model as a collection of parametric *model contexts*: $\Omega = \{\langle \omega_c, \vec{\varphi}_c \rangle\}_{c=1}^{C}$, where ω_c identifies the c-th model context and is associated with vector of parameters $\vec{\varphi}_c$.

In order to build such a model, we have to match homologous image contexts across images of our training set. Therefore, each model context ω_c is learned by using a training set $\mathscr{T}_c$ of CSBC descriptors matched across the different training images. In order to explain how the matching is performed, let us first explain how an image is recognized once we have learned the model.

3.1 Recognition

Suppose that we have learned a model $\Omega = \{\langle \omega_c, \vec{\varphi}_c \rangle\}_{c=1}^{C}$, and we get a new image I that we want to evaluate, i.e., decide whether or not it contains an object that is an instance of the model Ω. Assume for now that descriptors are only extracted at one scale, so that the image I is represented by only one set $H = \{\vec{h}_i\}_{i=1}^{R}$.

Let $l(\vec{h}|\omega_c) \in [0,1]$ be the likelihood that the contextual descriptor $\vec{h}$ represents the model context ω_c. This likelihood is based on learned parameters $\vec{\varphi}_c$, and we use Boosting as explained in section 4. Let $L(H|\omega_c) \in [0,1]$ be the likelihood that *any* contextual descriptor in H represents ω_c. For computing this likelihood we use the maximum: $L(H|\omega_c) = \max_{\vec{h}_i \in H} l(\vec{h}_i|\omega_c)$. This can be regarded as matching the model context ω_c with the contextual descriptor $\vec{h}_{i^*}$ whose likelihood is maximum. We express this as $M(H|\omega_c) = \vec{h}_{i^*}$, where $M(H|\omega_c) = \vec{h}_{i^*} = \arg\max_{\vec{h}_i \in H} l(\vec{h}_i|\omega_c)$. Let $\mathbf{L}(H|\Omega) \in [0,1]$ be the likelihood that H represents the object according to the evidence provided by all the model contexts $\{\omega_c\}_{c=1}^{C}$ of our model constellation. As we want all the model contexts to contribute to this classification score, we use as combination rule the sum of likelihoods, with equal weight for each model context: $\mathbf{L}(H|\Omega) = \frac{1}{C}\sum_{c=1}^{C} L(H|\omega_c)$.

Consider now multiple scales $\mathscr{H} = \{H_s\}_{s=1}^{S}$ for image I. Let $\mathscr{L}(\mathscr{H}|\Omega)$ be the probability that *any* of the scaled representations $H_s \in \mathscr{H}$ of image I contains our object. This is computed again by using the maximum $\mathscr{L}(\mathscr{H}|\Omega) = \max_{H_s \in \mathscr{H}} \mathbf{L}(H_s|\Omega)$. Again, this can be regarded as matching the model object Ω with some scaled representation H_{s^*}, which is expressed as: $\mathscr{M}(\mathscr{H}|\Omega) = H_{s^*} = \arg\max_{H_s \in \mathscr{H}} \mathbf{L}(H_s|\Omega)$. The described procedure is similar to multi-scale Chamfer matching [2], using learned likelihoods instead of distances. This matching can be efficiently performed if we exploit the sparseness of our representation by means of inverted files, which is explained in section 5.

3.2 Matching with low supervision

As explained above, before learning the model we match homologous contexts in the training set. Our procedure consists of two stages. In the first stage, we apply the registration procedure of [3] to a small set of manually segmented images. As a result of

registration, we obtain sets of homologous contexts across a small number of images, and we can learn an initial model Ω'. Then, we use this model and the matching explained in the recognition section (functions $M(H|\omega'_c)$ and $\mathscr{M}(\mathscr{H}|\Omega')$) to obtain homologous descriptors for the rest of the training set. Basically, given a non-segmented image I from the training set, we match every model context $\omega'_c \in \Omega'$ with the descriptor of image I that maximizes the likelihood of representing ω'_c, and we use the scale that has the highest likelihood according to all the model contexts $\{\omega'_c\}_{c=1}^{C}$. After this matching, we can gather the complete sets of homologous contexts for all the training set, and learn the final model $\Omega = \{\omega_c\}_{c=1}^{C}$. We refer to [1] for a more detailed description of the procedure.

In order to learn the initial model Ω', we use Generalized Correlograms (GCs) with quantized local descriptors that represent the direction of edges, as done in previous work [1]. The motivation is that the first matching produces small *training sets* of homologous contexts and it is better to utilize GCs that have a small number of dimensions. In order to obtain the final model Ω, we use as contextual descriptors a concatenation of the previous GCs and the Class-Specific Binary Correlograms (CSBC) described in previous section 2.

4 Learning a compact model by fast binary feature selection and Boosting

We want to obtain a final model that is compact, in the sense that it is based on few dimensions of the large feature space where CSBCs live. For this purpose we propose to use a feature selection algorithm especially efficient in binary spaces, based on maximizing the conditional mutual information between selected features and the class label, and introduced by Fleuret in [8]. This algorithm allows to efficiently perform feature selection in very large binary spaces, and obtains a small set of dimensions that convey a high amount of information about our object category. In our case, each dimension flags the presence of some class-specific local feature located in some relative spatial position. Therefore, we obtain a compact model characterizing the relevant local parts of the object and their characteristic mutual position.

The Conditional Mutual Information Maximization (CMIM) [8] works by selecting those features that maximize their mutual information with the object class, conditional to any feature already picked. It ensures the selection of features which are both individually informative and two-by-two weakly dependant [8]. After efficiently obtaining a small set of class-informative features by CMIM, we use Boosting with decision stumps to learn the final parametric contexts $\vec{\varphi}_c$ of section 3. We use the Boosting implementation described by Viola and Jones in [19], which can be easily adapted to work with sparse data, as in our case, in order to efficiently learn the model, we refer to [1] for further detail. Further, Boosting with weak classifiers allows to easily and efficiently work with Inverted Files in order to match the model with new images. This is explained below.

5 Efficient matching with Inverted Files

If we use the Chamfer matching algorithm described in 3.1, and we use Boosting with decision stumps, we can easily adapt the matching procedure to efficiently work with Inverted Files (IF) from the information retrieval community [17], as we will briefly explain in this section (for more details we refer to [1]). We describe here a procedure valid for

Method	Motorbike	Plane	Car rear	Face	Cat	Leaf
Others, Gray	92.5% [6]	90.2% [6]	90.3% [6]	**96.4**% [6]	90.0% [6]	84.0% [20]
GC [1], Gray	**96.7**%	92.5%	**95.8**%	95.2%	**92.8**%	**98.0%**
GC [1], Col.	94.2%	94.4%	95.8%	90.5%	86.1%	96.3%%
CSBC, 250 c.s., 2000 d., Col.	**98.3**%	**98.7**%	**99.6**%	**95.7**%	**89.0**%	**100**%
CSBC, 25 c.s., 250 d., Col.	97.5%	96.4%	98.7%	94.4%	80.1%	95.7%

Table 1: ROC equal error rate in CALTECH.

both binary CSBCs and non-binary GCs, because we use GCs in the first matching step of 3.2. Each decision stump from Boosting is defined by a threshold θ, a selected dimension k and a weight α. The stump evaluates as positive those descriptors whose value in k-th dimension is lower than θ. For each entry (dimension) in the IF the *negative* values are stored in ascending order, so that the virtual zeros are put at the end of the list (when training the classifier we also use negative values). From the first descriptor found, we visit all the descriptors until the beginning of the list and we add α to the likelihood of these descriptors. Let p_k be the fraction of descriptors in the list whose value is lower than θ: $0 \leq p_k \leq 1$. The cost of evaluating the weak classifier is $O(\log n_k + p_k n_k)$. Note that we can evaluate all the weak classifiers that share the same dimension k in only one pass along the IF list of this dimension.

6 Results

In order to test the performance of the method, we used the CALTECH database, which has become a standard database for object-class recognition. For a description of the database and image examples we refer to [6], we do not show them here due to lack of space. Most of the object categories have instances under the same bi-dimensional arrangement, except for the cat category. Each category has roughly 800 images of different objects of this category. From the positive training set, only 10 images are manually segmented and the rest are left unsegmented. Two background sets are used: the original gray-level backgrounds [6] and color bakgrounds. The latter was found to be harder to classify than the gray-level background set.

In table 1 we compare the performance against well-known geometry-based methods and against our previous work using Generalized Correlograms [1]. The first two rows are scores with Gray backgrounds and the three last rows use more difficult Color backgrounds. 250 c.s + 2000 d. means that we use 250 class-specific features and we select 2000 dimensions for obtaining the final model. We used 2000 dimensions for obtaining the highest scores, although a much smaller number (250) obtain similar scores. As we can see, CSBCs significantly outperform GCs with color bacgkrounds, and GCs in turn outperform the well-known benchmark of Fergus et al. [6] with gray-levels. We did not test our CSBCs with gray-level backgrounds. In table 2 we compare our method against well-known methods that also use dictionaries of class-specific features but without integrating geometry. We implemented the bag-of-feature methods by using histograms, as in [13]. For fair comparison, we used the same setting (local descriptors, vocabulary extraction, etc.) as the one used for our CSBCs. We also implemented the version of Opelt et al. [14] by using the same setting as in our method. We can see that using Class-Specific Binary Correlgrams we outperform methods based only on class-specific local features.

Method	Motorbike	Plane	Car rear	Face	Cat	Leaf
Bag-of-features, 250 c.s.	90.8%	95.0%	98.9%	95.0%	91.0%	96.7%
dist. vec. [14] 250 c.s.	96.0%	95.8%	99.1%	95.2%	**92.0**%	96.7%
CSBC, 250 c.s., 2000 dim.	**98.3**%	**98.7**%	**99.6**%	**95.7**%	89.0%	**100**%
dist. vec. [14] 25 c.s.	90.0%	95.0%	95.6%	91.7%	**88.0**%	94.6%
CSBC, 25 c.s., 250 dim.	**97.5**%	**96.4**%	**98.7**%	**94.4**%	80.1%	**95.7**%

Table 2: ROC equal error rate in CALTECH. Comparison with methods based on dictionaries of class-specific features.

Using CSBCs can be seen as a post-processing step after performing a method that obtains class-specific features, such as [13, 14]. The cost of extracting the multi-scale CSBCs in a Pentium IV at 3.0GHz was around 0.7 seconds per image, with code implemented in Matlab and some subroutines in C. The cost of classifying an image with 2000 selected features and inverted file was around 1/3 seconds in a Pentium IV at 3.0 GHz. Note that a smaller number of dimensions is enough for obtaining almost the same results: with 500 dimensions the average difference in accuracy is 0.62%, and we classify an image in $1/15$ seconds. Learning CSBCs of 15000 dimensions and 2000 weak classifiers takes about 80 minutes, whereas using 500 dimensions we spend only 20 minutes approximately. These costs are much smaller than using fast graph-based methods such as [6], were the authors spend around 24 hours (in a 2.0 GHz processor). Using our GCs in [1] and a 2.4 GHz we spent 1 hour and 30 minutes for the whole learning algorithm and 250 selected features.

CSBCs share the same robustness as GCs, evaluated in [1]. Notably they are invariant to translation, robust against scaling in our case (we use a multi-scale approach) and robust against small rotations. This is in contrast with recently proposed fast geometry-based methods that are not robust against pose variation.

7 Conclusions

We showed how to efficiently consider spatial relations by using Class-Specific Binary Correlograms. For this purpose, we outlined an architecture were all the components are designed so as to efficiently learn and match this type of descriptors, i.e. by exploiting their sparseness and their binary nature. Notably, we saw that the proposed method outperforms recent methods based on dictionaries of class-specific features. In this sense, using CSBCs can be considered as an efficient post-processing step after after obtaining a set of class-specific features used in many recent works [13, 14, 12, 16]. We saw that this step can be done without high cost and significantly increments performance. As future work we will test it in the CALTECH-101 database and will use rich features as [12, 16].

References

[1] J. Amores, N. Sebe, and P. Radeva. Context-based object-class recognition and retrieval by generalized correlograms. *PAMI*. IN PRESS (on-line at IEEE web site).

[2] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proc. Int. Joint Conf. Artificial Intelligence*, pages 659–663, 1977.

[3] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 24(24):509–522, April 2002.

[4] D. Crandall, P. Felzenszwalb, and D. Huttenlocher. Spatial priors for part-based recognition using statistical models. In *IEEE CVPR*, 2005.

[5] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *Proc. CVPR*, 2005.

[6] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. CVPR*, 2003.

[7] R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. In *Proc. CVPR*, pages 380–387, 2005.

[8] F. Fleuret. Fast binary feature selection with conditional mutual information. *Machine Learning*, 5:1531–1555, 2004.

[9] P. Hong and T. S. Huang. Spatial pattern discovery by learning a probabilistic parametric model from multiple attributed relational graphs. *Journal of Discrete Applied Mathematics*, 139(1-3):113–135, April 2003.

[10] J. Huang, S. R. Kumar, M. Mitra, W. J. Zhu, and R. Zabih. Spatial color indexing and applications. *IJCV*, 35(3):245–268, 1999.

[11] T. Kadir and M. Brady. Scale, saliency and image description. *IJCV*, 2001.

[12] J. Mutch and D. Lowe. Multiclass object recognition with sparse, localized features. In *CVPR*, volume 1, pages 11–18, 2006.

[13] E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In *Proc. ECCV*, 2006.

[14] A. Opelt, A. Pinz, M. Fussenegger, and P. Auer. Generic object recognition with boosting. *PAMI*, 28(3):416–431, 2006.

[15] S. Lazebnik C. Schmid and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. CVPR*, 2006.

[16] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. Robust object recognition with cortex-like mechanisms. *PAMI*, 29(3):411–426, 2007.

[17] D. Squire, W. Muller, H. Muller, and T. Pun. Content-based query of image databases: Inspirations from text retrieval. *Pattern Recognition Letters*, 2000.

[18] S. Ullman, M. Vidal-Naquet, and E. Sali. Visual features of intermediate complexity and their use in classification. *Nature neuroscience*, 5(7):682–687, 2002.

[19] P. Viola and M. J. Jones. Robust-real time face detection. *IJCV*, 2004.

[20] M. Weber, M. Welling, and P. Perona. Towards automatic discovery of object categories. In *CVPR*, pages 101–108, 2000.

[21] J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. In *Proc. ICCV*, 2005.

Generalised Linear Pose Estimation

Andreas Ess, Alexander Neubeck, Luc Van Gool
Computer Vision Lab
ETH Zurich
8092 Zurich, Switzerland
{aess, aneubeck, vangool}@vision.ee.ethz.ch
http://www.vision.ee.ethz.ch

Abstract

This paper investigates several aspects of 3D-2D camera pose estimation, aimed at robot navigation in poorly-textured scenes. The major contribution is a fast, linear algorithm for the general case with six or more points. We show how to specialise this to work with only four or five points, which is of utmost importance in a test and hypothesis framework. Our formulation allows for an easy inclusion of lines, as well as the handling of other camera geometries, such as stereo rigs. We also treat the special case of planar motion, a valid restriction for most indoor environments. We conclude the paper with extensive simulated tests and a real test case, which substantiate the algorithm's usability for our application domain.

1 Introduction

Given known points in the world and their projected images, pose estimation refers to recovering the pose of the camera from such data. In photogrammetry, this problem has been studied as early as 1841 by Grunert [8]. For a recent survey, consult e.g. [9].

We are interested in pose estimation for the purpose of navigating a robot with a stereo rig through a poorly-textured room. Besides the usual demands on speed and accuracy, a pose estimation algorithm should use as few feature points as possible during robust estimation in such a setting. To ease the problem of lacking feature points, it is advantageous to allow other features besides points, such as lines, in the estimation. Moreover, the algorithm should be extendable to a stereo rig. Lastly, as the robot movements might be restricted to planar ones, it should be possible to include this constraint in the formulation. All these demands are fulfilled by our algorithm: four points suffice for the estimation in the general case, we show how lines can be used in conjunction with points, we investigate the special case of planar motion, and demonstrate the application of the algorithm to a stereo rig in this planar case. While the presented results are for a robotic scenario, the algorithm is also directly applicable to e.g. augmented reality.

Before presenting the actual algorithm in Section 2, the next subsection lists some existing solutions to the general problem of linear pose estimation. Section 3 investigates the specialisation of the algorithm for the planar case. Then, Section 4 gives results from both simulated and real-world scenarios, before the paper is concluded in Section 5.

1.1 Related Work

A linearisation of the pose estimation problem is not only tempting for reasons of efficiency. An accurate linear solution also provides a valuable starting point for an iterative algorithm, resulting in its faster convergence. First, we discuss three prior contributions to linearising the n-point pose estimation problem.

The first linear approach for determining the camera pose was proposed by Fiore [6]. Based on a projective invariant, he constructs a linear equation system to solve for the depths of the image points. Using the recovered depths, 3D-3D pose estimation (also called absolute orientation, [10]) is carried out to obtain the actual pose. In the general case, the method needs at least six point correspondences to work.

Based on a general procedure for linearising quadratic systems, Ansar and Daniilidis [1] suggest methods for pose estimation from points and lines. For points, their approach is based on depth recovery by considering ratios between the points' depths. For lines, their respective angles and the invariance of inner and outer products are used to directly estimate the pose. In both cases, two linear systems are constructed to guarantee a unique solution. While mathematically interesting, this approach mainly suffers from high-dimensional equation systems, which considerably hamper the performance of the algorithm, both in terms of speed and numerical accuracy.

Quan and Lan [15] also base their algorithm for depth recovery on ratios between different points' depths. They apply their method to each point in turn and solve a set of fourth degree polynomial constraints. Their algorithm works for $n \geq 4$ points.

All the above methods need an additional step of absolute orientation. The algorithm proposed in this paper recovers the pose directly. It works for both points ($n \geq 4$) and lines ($n \geq 5$), and a mixed solution is possible. As shown in the results section, is performance is comparable with or better than other algorithms.

As a note on minimal solutions, Nister [12] presents a generalised solution for the 3-point case (an additional point is needed for resolving a 4-fold ambiguity). Besides minimality, an advantage of the method is its applicability to stereo rigs or other generalised camera systems by allowing non-concurrent rays. The algorithm presented here offers the same generality, but also allows more points or the usage of lines.

Phong *et al.* [14] also present an algorithm for calculating the pose from points and lines, however, their solution is iterative. Liu *et al.* [11] study a two-step process for pose estimation from lines that first iteratively solves for rotation and then linearly for translation. Point tuplets can be included as lines. They also present a linear solution for the first step, which however is not minimal in the number of needed correspondences, as opposed to the one presented here. Also, no special cases (planar motion, multiple cameras) are studied.

2 Pose from Points

Given a set of world points $\mathbf{p}_i$ and a corresponding set of *normalised*[1] image points $\mathbf{q}_i$, the pose estimation problem consists of finding a rotation matrix $\mathtt{R}$ and a translation vector $\mathbf{t}$ that minimise the geometric reprojection error:

$$\sum_i \|\mathtt{R}\mathbf{p}_i + \mathbf{t} - \lambda_i \mathbf{q_i}\| / \lambda_i \rightarrow min. \tag{1}$$

[1] For actual points $\tilde{\mathbf{q}}$ in the image, $\mathbf{q} = \mathtt{K}^{-1}\tilde{\mathbf{q}}$, for lines $\tilde{\mathbf{n}}$, $\mathbf{n} = \mathtt{K}^{\top}\tilde{\mathbf{n}}$, with $\mathtt{K}$ the internal calibration matrix.

In Eq. (1), λ_i models the depth of a point from the given view. Since this is a non-linear problem in λ_i, iterative algorithms are needed for the solution. However, instead of considering the geometric reprojection error, one could minimise the distance between the two back-projected rays, also known as object space collinearity:

$$\sum_i \|\mathtt{R}\mathbf{p}_i + \mathbf{t} - \lambda_i \mathbf{q_i}\| \rightarrow min. \tag{2}$$

While this formulation does not take the depth-dependent uncertainty of the 3D location into account, it allows for a linear, and hence fast and simple, solution to the problem. By setting the derivatives of λ_i to zero, the λ_i can be eliminated. Let $\mathtt{Q}_i = \mathtt{I} - \frac{\mathbf{q_i}\mathbf{q_i}^\top}{\|\mathbf{q_i}\|}$ denote the projection operator into $\mathbf{q_i}$. After some rearranging, Eq. (2) can be reformulated as

$$\sum_i \|\mathtt{Q}_i(\mathtt{R}\mathbf{p}_i + \mathbf{t})\| \rightarrow min. \tag{3}$$

Assuming an affine transformation instead of $\mathtt{R}$, Eq. (3) becomes a linear system $\mathtt{M}\mathbf{x} = \mathbf{0}$ with twelve unknowns whose least squares solution is wanted. This can be solved in a straightforward manner. However, as the values of $\mathtt{R}$ are between -1 and 1, while the ones for $\mathbf{t}$ can take any value, $\mathbf{t}$'s variables can dominate the solution vector, resulting in unsatisfactory precision for $\mathtt{R}$. It is not possible to impose the hard-constraints of orthonormality on $\mathtt{R}$ in a linear fashion due to their quadratic nature. Still, the less restrictive constraint $\|\mathtt{R}\|_F = \sqrt{3}$, where $\|\cdot\|_F$ is the Frobenius norm, can be enforced, as shown in the following. The above problem is partitioned as follows:

$$(\mathtt{A}\ \mathtt{B}) \begin{pmatrix} \mathbf{r} \\ \mathbf{t} \end{pmatrix} = \mathbf{0}, \tag{4}$$

where $\mathbf{r}$ corresponds to the nine variables of the rotation, and $\mathtt{A}$ and $\mathtt{B}$ to their respective columns in $\mathtt{M}$. To find a least squares solution, we square Eq. (4):

$$\begin{pmatrix} \mathbf{r}^\top & \mathbf{t}^\top \end{pmatrix} (\mathtt{A}\ \mathtt{B})^\top (\mathtt{A}\ \mathtt{B}) \begin{pmatrix} \mathbf{r} \\ \mathbf{t} \end{pmatrix} \rightarrow min. \tag{5}$$

Deriving of the term for $\mathbf{t}$ and setting to zero yields

$$\mathbf{t} = -(\mathtt{B}^\top\mathtt{B})^+\mathtt{B}^\top\mathtt{A}\mathbf{r} = -\mathtt{B}^+\mathtt{A}\mathbf{r}, \tag{6}$$

where $\mathtt{B}^+$ denotes the pseudo-inverse of the matrix $\mathtt{B}$. Eq. (4) can now be rewritten as

$$\mathtt{A}\mathbf{r} - \mathtt{B}\mathtt{B}^+\mathtt{A}\mathbf{r} = 0. \tag{7}$$

Considering the economy-size SVD[2] of $\mathtt{B} = \mathtt{U}\mathtt{S}\mathtt{V}^\top$, this simplifies to

$$(\mathtt{A} - \mathtt{U}\mathtt{U}^\top\mathtt{A})\mathbf{r} = 0. \tag{8}$$

Eq. (8) is a least squares problem in the rotation $\mathbf{r}$, whose solution can be found as the nullvector of SVD. By definition of the SVD, the resulting vector $\mathbf{r}$ will have length 1. Hence, a multiplication by $\sqrt{3}$ gives the desired Frobenius norm for the rotation. $\mathbf{t}$ is then obtained by backsubstitution into Eq. (6).

As we are basically only solving for an affine transformation, it is advisable to orthogonalise the obtained solution for $\mathtt{R}$ *before* solving for $\mathbf{t}$. Compared to QR decomposition and quaternions, SVD yielded the best results for this (QR and quaternions were both about 10% worse in average w.r.t. accuracy). The SVD might return an orthogonal $\mathtt{R}$ with $\det(\mathtt{R}) = -1$, which can be fixed with a sign reversal of both $\mathtt{R}$ and $\mathbf{t}$.

[2]The economy-size SVD of an $m \times n$ matrix yields an $m \times n$ matrix $\mathtt{U}$, containing only the columns corresponding to non-zero singular values. $\mathtt{U}\mathtt{U}^\top$ consequently does *not* correspond to the identity. $\mathtt{S}$ and $\mathtt{V}$ are $n \times n$.

2.1 4 or 5 Points

As each point correspondence imposes only two constraints, a minimum of six point correspondences is needed to solve the pose estimation problem uniquely. For less points, the dimension N of the nullspace of Eq. (8) will not be 1 (disregarding degenerate cases, $N = 12 - 2n$). Let's denote the nullspace of Eq. (8) by

$$\mathbf{X} = \{\mathbf{r}_1, \ldots, \mathbf{r}_N\} \tag{9}$$

The correct solution $\mathbf{r}$ is a linear combination of the vectors $\mathbf{r}_i$. That is, we need to find scalars α_i, such that

$$\mathbf{r} = \sum_i \alpha_i \mathbf{r}_i. \tag{10}$$

To obtain this, a quadratic system based on the constraints $\mathtt{R}^\top\mathtt{R} = \mathtt{R}\mathtt{R}^\top = \mathtt{I}$ can be constructed. These quadratic constraints on $\mathtt{R}$ are formulated in the quadratic terms of its elements, $r^{(i)}r^{(j)}$, where $r^{(i)}$ is given by Eq. (10). Now, we linearise the resulting quadratic terms in $\alpha_i\alpha_j$ by substituting them with new variables α_{ij}. Due to the linearised nature of the formulation, there are eleven independent constraints on the r_{ij}, and hence on the α_{ij}. Thereby, we obtain a linear equation system in the $\frac{N(N+1)}{2}$ scalars α_{ij}. For five points, this corresponds to three unknowns, for four points, to ten unknowns. The α_i can be recovered up to a global sign ambiguity from the α_{ii}. As above, this ambiguity is resolved by enforcing $\det(\mathtt{R}) = 1$.

2.2 Lines

The chosen formulation also allows for the direct inclusion of lines (or other features, for that matter) into the pose estimation. In contrast to previous work [1], both lines and points can be used at the same time to constrain the pose. A line's direction imposes the following constraint on $\mathtt{R}$:

$$\mathbf{n}^\top\mathtt{R}\mathbf{d} = 0, \tag{11}$$

with $\mathbf{d}$ being the unit vector of the 3D line's direction and $\mathbf{n} = (a,b,c)^\top$ the normalised line in 2D. $\mathbf{n}$ is set to unit length as well, and equals the normal vector of the plane obtained from the 2D line's back-projection. In this formulation—which proved to be the most stable one in our experiments—each line imposes only one constraint on $\mathtt{R}$. Still, using the steps from subsection 2.1, five instead of nine lines suffice to estimate the pose.

2.3 Multiple Cameras

When the joint pose estimation from multiple, externally calibrated cameras is desired, the right hand side of Eq. (4) is not $\mathbf{0}$ anymore. The solution to the resulting constrained problem $\mathtt{A}\mathbf{x} = \mathbf{b}, \|\mathbf{x}\| = 1$ then can be found either using differential-geometry methods [4], or by introducing a homogeneous variable, as done for planar motion in Section 3. As latter disables the direct constraint $\|R\|_F = 3$, former is to be preferred.

2.4 Depth Dependency

Clearly, using object-space collinearity instead of the more correct image-space collinearity seems like a disadvantage. Please note however that the depth λ_i is mainly a weighting

factor in Eq. (1), for which in many cases an estimate is available. In Section 4, we also examine the algorithm's performance subject to different depth ranges.

3 Planar Motion

The special case of robot navigation in indoor environments typically restricts the movements to planar ones. This greatly reduces the number of unknowns and can thereby increase stability and performance. There has been quite some research activity in this field, mostly in the uncalibrated case, cf. e.g. [2, 5]. Stewenius *et al.* [16] treat various problems for multi-camera setups subject to planar motion. Their resection algorithm is similar to the one presented in the following, was however very susceptible to noise in our tests.

In the planar case, the motion variables reduce to

$$\mathtt{R} = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix}, \quad \mathbf{t} = \begin{pmatrix} t_x \\ 0 \\ t_z \end{pmatrix} \tag{12}$$

Inserting into Eq. (3), this yields equations of the form

$$(Q_{11}p_x + Q_{31}p_z)c + (-Q_{31}p_x + Q_{11}p_z)s + (1-Q_{11})t_x + (-Q_{13})t_z = -Q_{21} \tag{13}$$

$$(Q_{13}p_x + Q_{33}p_z)c + (-Q_{33}p_x + Q_{13}p_z)s + (Q_{31})t_x + (1-Q_{33})t_z = -Q_{23}, \tag{14}$$

where Q_{ij} corresponds to the entries of the projection matrix $\mathtt{Q}$. Besides the necessary variables for $c = \cos\theta$ and $s = \sin\theta$, we introduce an additional homogeneous variable ρ for Q_{21} and Q_{23} to arrive at a formulation needed for Eq. (4). After having solved for $\mathbf{r} = (c', s', \rho)^\top$ in Eq. (8), we remove the homogeneous quantity by dividing through ρ. A valid rotation is then inferred from c and s.

In the reduced formulation, three points suffice to determine the pose using Eq. (8) and Eq. (6) without any further processing. Vertical lines can be treated as points when considering their intersection with a horizontal plane. Also, due to the introduction of the homogeneous variable, camera rigs can be handled in a straight-forward fashion.

4 Results

In the following, we will give various simulation results of our algorithm ("GLPE") compared to the approaches suggested by [1] ("Ansar") and [6] ("Fiore"), as well as the solution obtained using Levenberg-Marquardt ("LM", initialised using GLPE). Due to its iterative nature, the latter is expected to always come closest to the actual minimum; we include it mainly as a reference. In all cases, the rotational error is measured as the absolute error in the unit quaternion, $|q - q_0|$. The relative translational error is measured as $\frac{2\|t-t_0\|_2}{\|t\|_2+\|t_0\|_2}$. q_0 and t_0 refer to the ground truth for rotation and translation, respectively.

4.1 Simulation

All the results in this section were obtained by averaging over 5,000 runs for the respective scenario. Different configurations for the point clouds (different sizes and shapes, such

as pyramids and cubes) were tested, but the basic results with respect to accuracy and the relative ordering among the algorithms remained the same. Thus, we only report the results that were obtained by randomly sampling points in a $10,000 \times 10,000 \times 10,000$ cube, centered around the z-axis, and placed in front of a camera in the origin. The point cloud's barycentre is then assumed as new origin, around which the entire system is rotated randomly. A focal length of $1,500$ was chosen, with principal point $\mathbf{0}$, and aspect ratio 1. Gaussian noise is added to the projected image points.

Fig. 1 shows the results obtained by keeping the number of points constant at six and increasing the noise level. For the rotational error, our algorithm clearly outperforms the other linear approaches. The error in translation is comparable between the three algorithms. As an interesting sidenote, Fiore performs very comparably to Ansar, as opposed to the findings by [1].

Next, we compare the performance with respect to the number of points used, Fig. 2. We perform slightly worse than Ansar for the 4-point case, as only eleven constraints are imposed on the ten unknowns, cf. Section 2.1. The 5-point case exhibits a drastic gain in accuracy and outperforms Ansar's algorithm. There is another considerable improvement for more than six points, as this presents the overdetermined case. Note that Fiore's approach could not be used with less than six points. In general, it performs rather similar to our algorithm. However, besides offering various useful extensions (lines, planar motion), we found our algorithm to be more reliable in a real scenario, as shown later.

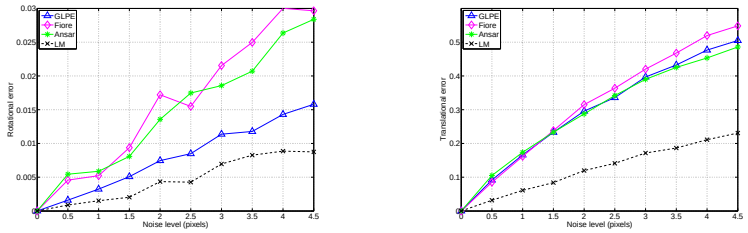


Figure 1: Rotational and translational error with increasing Gaussian noise (6 points).

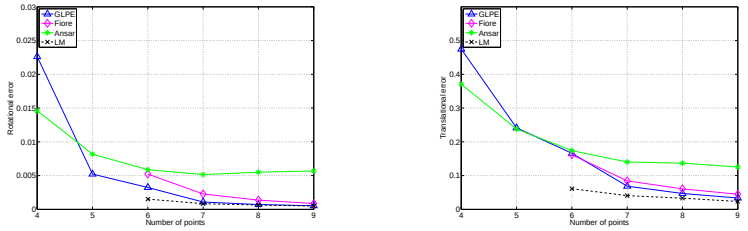


Figure 2: Performance with increasing number of points (noise $\sigma = 1.5$ pixels).

For lines, we compare our algorithm to the line algorithm proposed in [1], Fig. 3. While our formulation needs at least five lines (as opposed to four), it is more accurate in all tests. Also, keep in mind that our formulation would allow for a mixed solution, which is not possible using their approach.

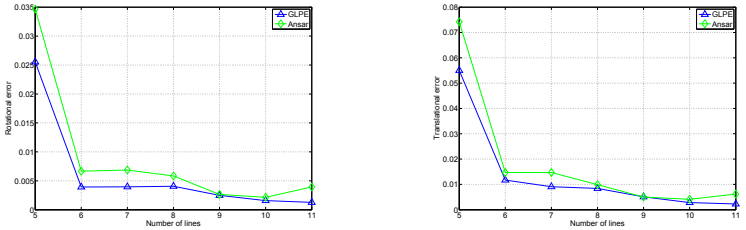


Figure 3: Performance with increasing number of lines (noise $\sigma = 1.5$ pixels).

4.2 Depth Dependency

Lastly, we investigate the performance of the various algorithms by increasing the depth range where points can come from, Figure 4. Again, we use 6 points and a noise standard deviation of 1.5 pixels. As can be seen, the rotational error is largely unaffected by increasing depth range, however, translational error grows. As mentioned above, in most cases, a depth estimate is known for the points (especially when processing video sequences), which can be used to weight the influence of each point.

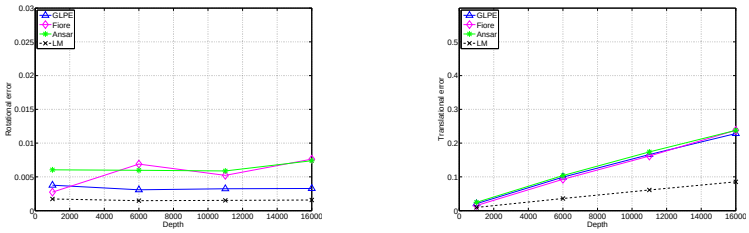


Figure 4: Performance with increasing depth range (6 points, noise $\sigma = 1.5$).

4.2.1 Planar Motion

For the planar case, we compare our algorithm with the one of Stewenius *et al.* [16]. Obviously, it is not quite fair to compare any of the above algorithms with the special case of planar movement. Nonetheless, for illustrative purposes, we add Fiore's algorithm to the comparison. Fig. 5 and Fig. 6 again give results for varying noise level and number of points, respectively. As to be expected, our algorithm outperforms Fiore for the same number of points, especially when noise levels are increasing. Notice however that the planar version already outperforms Fiore given three points less, which is basically the number of missing degrees of freedom. The algorithm of Stewenius proved to be very susceptible to noise in our experiments and was hence left out in Fig. 6.

4.3 Timings

We time straightforward C-implementations of the aforementioned algorithms, using VXL for the numerical routines (everything was compiled *without* optimisations). Table 4.3 gives an overview of the runtimes in ms on a 2.4 GHz AMD Opteron for various numbers

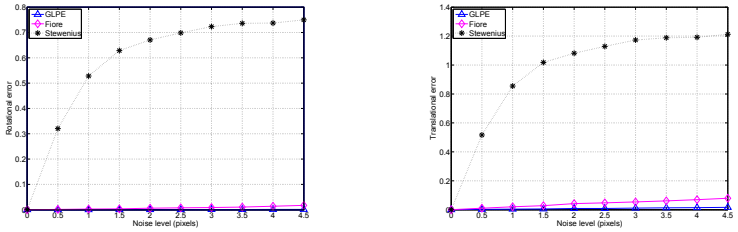


Figure 5: Performance with increasing Gaussian noise in the planar case (6 points).

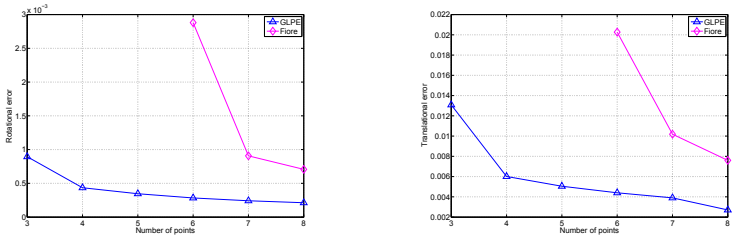


Figure 6: Performance with increasing number of points in the planar case.

of points, averaged over 10,000 runs. As expected, our algorithm has very low runtimes due to its small equation system. In the 4- and 5-point cases, some additional overhead is needed for finding the correct linear combination in the nullspace (see section 2.1), hence the slightly higher runtime. The size of this second equation system decreases enough in the 5-point case to yield attractive timing figures. Fiore provides an overall fast performance, but is slightly slower than our algorithm. With an increasing number of points, the margin between Fiore and our algorithm increases (note however that Fiore proposes an optimisation that is not implemented here). Using many points at low cost in time is attractive for providing a better initialisation using all inliers, and hence quicker convergence, for a nonlinear algorithm after running RANSAC [7]. Ansar's algorithm becomes very slow for an increasing number of points, as the size of the equation system increases quadratically as opposed to linearly as in our and Fiore's formulation.

The planar version takes 0.060 ms for the 4-point and 0.086 ms for the 8-point case.

Algorithm	4 points	5 points	6 points	8 points	50 points	100 points
GLPE	0.346	0.205	0.194	0.227	0.865	1.657
Ansar	0.727	2.016	5.159	28.031	-	-
Fiore	-	-	0.201	0.311	9.362	38.590

Table 1: Runtimes in ms for different algorithms on a 2.4 GHz AMD Opteron.

4.4 Real Scenario

To verify the results obtained in the above simulations, we apply the presented method to a small video sequence taken from a stereo rig, which undergoes a planar motion. Two

moving persons are walking in front of the robot, introducing an additional challenge. In all images, SURF features [3] are detected. Similar to [13], an initial triangulation from the stereo rig is used for the world reference points. The pose is estimated using either Fiore's, Ansar's, or our approach with 6 points, or our planar stereo 4-point algorithm. At every iteration, all inliers are used to iteratively refine the pose. Every fifth frame, the world points are discarded and new ones are triangulated from the current stereo frame. A few sample frames, as well as the obtained tracks are shown in Fig. 7. Even without using any bundle adjustment, our algorithms yield better results than using either Fiore's or Ansar's method. Already the 5-point version gives an acceptable result. The planar version, with sampling from both cameras of the stereo rig, exposes its advantages in this constrained scenario: while the other methods show quite some variance in y-direction (up to 20cm, not visible in the figures), its track sticks nicely to the ground. For comparison purposes, we also include the track generated using the 3-point algorithm and bundle adjustment.

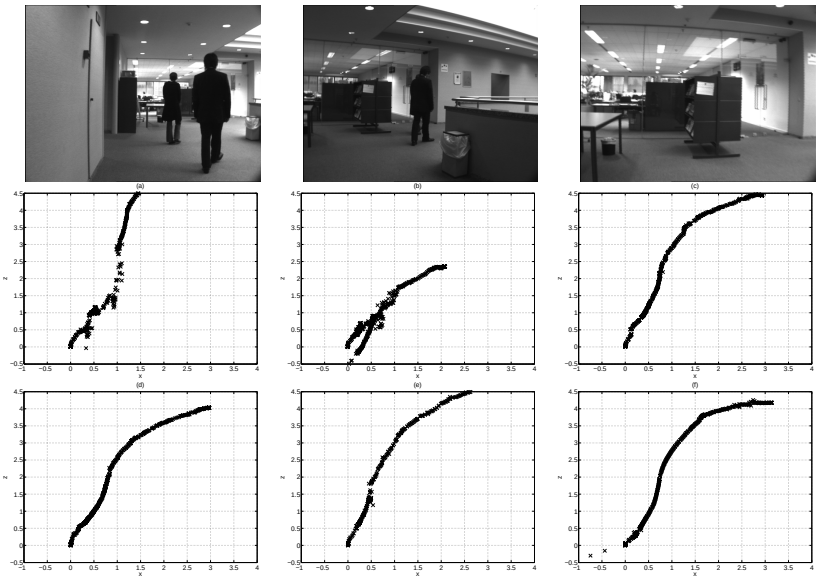


Figure 7: Real scenario. Top: frames with two moving objects. Middle: tracks found by 6-point versions of Fiore (a), Ansar (b) and GLPE (c). Bottom: 5-point version of GLPE (d), stereo planar GLPE 4-point (e), and 3-point relative pose with bundle adjustment (f).

As mentioned in the introduction, the algorithm can be applied to other scenarios such as augmented reality. In general, a better performance can be expected in cases where e.g. a 3D CAD model is already given, since this omits additional errors from 3D reconstruction as in the example shown here.

5 Conclusions

We introduced a novel algorithm for linear pose estimation from $n \geq 4$ points. Aimed at robot navigation in poorly-textured indoor environments, the algorithm's main advantage is its adaptability to different scenarios such as planar movement, stereo rigs or the inclusion of lines and other features besides points. Extensive simulation results show its general performance to be comparable or slightly better than other linear algorithms, at increased speed. In a real scenario, the algorithm outperforms other linear approaches, even without making use of advantages such as the support of stereo rigs or the ability to constrain the motion.

Acknowledgements

The authors acknowledge the support of Toyota Motor Europe and Toyota Motor Corporation. They would like to thank Kostas Daniilidis for providing the Matlab source code for their point algorithm.

References

[1] A. Ansar and K. Daniilidis. Linear pose estimation from points or lines. *PAMI*, 25(5):578–589, 2003.

[2] M. Armstrong, A. Zisserman, and R. I. Hartley. Self-calibration from image triplets. In *ECCV (1)*, pages 3–16, 1996.

[3] H. Bay, T. Tuytelaars, and L. van Gool. Surf: Speeded up robust features. In *ECCV*, 2006.

[4] L. Elden. Solving quadratically constrained least squares problems using a differential-geometric approach. *BIT*, 42:323–335, 2002.

[5] O.D. Faugeras, L. Quan, and P.F. Sturm. Self-calibration of a 1d projective camera and its application to the self-calibration of a 2d projective camera. *PAMI*, 22(10):1179–1185, 2000.

[6] P.D. Fiore. Efficient linear solution of exterior orientation. *PAMI*, 23(2):140–148, 2001.

[7] M.A. Fischler and R.C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6), 1981.

[8] J.A. Grunert. Das pothenotische problem in erweiterter gestalt nebst über seine anwendungen in der geodäsie. *Grunerts Archiv für Mathematik und Physik, Band 1*, pages 238–248, 1841.

[9] R. M. Haralick, C. Lee, K. Ottenberg, and M. Noelle. Review and analysis of solutions of the three point perspective pose estimation problem. *IJCV*, 13(3):331–356, 1994.

[10] B. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of Optical Society of America A*, Vol 4., No. 4:629–642, 1987.

[11] Y. Liu, T. S. Huang, and O. D. Faugeras. Determination of camera location from 2-d to 3-d line and point correspondences. *PAMI*, 12(1):28–37, 1990.

[12] D. Nistér. A minimal solution to the generalised 3-point pose problem. In *CVPR (1)*, 2004.

[13] D. Nistér, O. Naroditsky, and J. R. Bergen. Visual odometry. In *CVPR (1)*, 2004.

[14] T.Q. Phong, R. Horaud, A. Yassine, and P.D. Tao. Object pose from 2-d to 3-d point and line correspondences. *IJCV*, 15:225–243, 1995.

[15] L. Quan and Z.-D. Lan. Linear n-point camera pose determination. *PAMI*, 21(8):774–780, 1999.

[16] H. Stewénius, M. Oskarsson, and K. Åström. Reconstruction from planar motion image sequences with applications for autonomous vehicles. In *SCIA*, pages 609–618, 2005.

Overcoming Parallax and Sampling Density Issues in Image Mosaicing of Non-Planar Scenes

Zhi Qi
Centre for Intelligent Machines
McGill University
qizhi@cim.mcgill.ca

Jeremy R. Cooperstock
Centre for Intelligent Machines
McGill University
jer@cim.mcgill.ca

Abstract

Image mosaicing constructs a wide field-of-view result from multiple source frames. In order to ensure a perceptually correct result, mosaicing typically requires either a planar or near-planar scene, parallax-free camera motion between source frames, or a dense sampling of the scene. When these conditions are not satisfied, various artifacts may result. A novel mosaicing approach is introduced that overcomes these limitations, building on the techniques of image-based rendering and manifold mosaicing, while permitting the synthesis of an effective mosaic of a non-planar scene from a sparse set of translated cameras. Our method first generates a series of intermediate virtual frames to reduce the disparities between neighboring images. Next, a series of vertical slices are chosen from the array of both real and virtual frames and connected according to a cost function that maximizes the similarity between adjacent slices. Experimental results indicate significant improvements over competing methods.

1 Introduction

In traditional imaging, with a fixed camera sensor, one must choose between capturing a wide field of view or high resolution details. *Image mosaicing* combines multiple input images, typically with some overlap, to produce an output with both wide field of view and high resolution. In general, this assumes that the camera motion is strictly rotational, as in QuicktimeVR [6], or that the scene is limited in depth variance, i.e. is planar or nearly planar. Otherwise, parallax effects result in the same point appearing at different relative positions on the multiple camera images. Such artifacts of duplication and missing objects can be avoided either by ensuring a parallax-free camera configuration or by increasing the sampling rate [13][15][16] of the scene.

Our motivation for investigating this problem stems from our three-screen videoconferencing configuration, illustrated in Figure 1a, in which a camera above each screen supplies video to a corresponding display at the remote location. At present, we have to restrict users to sit in the non-overlapping regions between cameras and require a uniform background to prevent the same objects appearing duplicated across multiple screens. However, it would be highly desirable to relax these constraints through the use of image mosaicing, which tolerates parallax effects thus is capable to generate reasonable

panorama view of the scene contents visible to the ensemble of cameras, without concern for the depth at which their respective fields of view begin to overlap.

We considered the simpler possibility of placing all three cameras in close proximity over the center screen,[1] rotated with respect to one another, as in Figure 1*b*. In this case, no restriction is necessary in the maximal depth at which users may sit from the cameras because the rotational camera configuration is insensitive to depth variance at all. However, unfortunately, such a configuration biases the correct view perspective to those users sitting near the center. Observers who are sitting at the sides would receive incorrect directional cues thus, precluding effective face-to-face interaction between these individuals sitting "across the table" from each other. An additional reason for preferring a translated camera configuration is that this provides depth cues that can be used later to generate a stereoscopic rendering for immersive applications.

Hence, the challenge we face is to generate a reasonable mosaic from a sparse sampling of the environment, in which depth variance of objects of interest may be significant, and where the camera configuration does not avoid parallax. To reduce the effect of parallax, our solution first synthesizes a number of virtual images along the path between input cameras and then combines these, along with the actual input images, using *manifold mosaicing*[9][13], which is robust to non-parallax-free camera configurations when provided a sufficiently dense sampling of the scene.

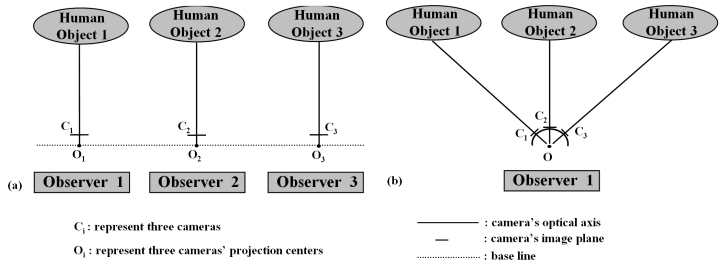


Figure 1: Two possible multi-screen videoconferencing camera configurations: (a) translational, which supports face-to-face communication across the table but suffers from overlapping fields of view; (b) rotational, which avoids problems of overlap but biases the perspective toward the center position.

Our work overcomes the conventional constraints of parallax-free camera motion or limited depth variance. Building on the techniques of image-based rendering and manifold image mosaicing, we further dispense with the need for dense sampling in the case of non-parallax-free camera motion. The result is an effective approach to image mosaicing where traditional mosaicing algorithms fail. The computational requirements of the algorithm we present here preclude real-time operation at video frame rates, but this should be overcome by taking advantage of the power of the programmable GPU. Even in its current implementation, we believe this represents an important step in the right direction.

The remainder of this paper is organized as follows. Previous work in image mosaicing is summarized in Section 2, followed by a detailed description of our approach in Section 3. The results of our algorithm, applied to various image databases, are presented

[1] This is the configuration favored by Cisco in their high-definition *Telepresenc their fiele* system.

in Section 4. Experimental results demonstrate that our system makes significant progress in mosaicing quality over *Autostitch* [4], in particular with respect to a reduction of ghost errors caused by parallax. Finally, possible improvements to our system are discussed in Section 5.

2 Previous Work

Traditional image mosaicing techniques include three basic steps. First, image registration is applied to find the geometric relationship between input images. Second, images are warped so that their regions of overlap match each other. Third, these warped images are stitched together into a common mosaicing plane.

The Concentric Mosaic [18] is an image-based rendering technique that constrains camera movement to planar concentric circles and creates panoramas by composting input images taken at different positions onto these circles. Shum and Szeliski's global and local alignment [19] greatly improved the accuracy of image registration without the prior known camera motion models so that tremendously enhanced the quality of mosaicing result from images taken by hand held cameras. Brown and Lowe [4] presented a fast image mosaicing system using robust feature-based image alignment, considered by many to be among the best algorithms currently available.[2] Levin *et al.*[11] investigated seamless image stitching in the gradient domain, which overcomes both the problems of photometric inconsistency and geometric misalignment between input images. In order to obtain reasonable mosaicing results, all of these systems require either the camera motion model to be parallax-free or the scene to exhibit little depth variance.

Another approach known as manifold or strip mosaicing [9][13][14], generates a multiperspective panorama by projecting thin strips from input images onto the mosaicing plane. The shape of the strips depends on the motion of input cameras and the width is proportional to the amount of camera motion.

Rav-Acha [15] addressed the problem of more complex camera motion in 3*D* space by time warping, which resamples the time axis to generate straight feature line in EPI space and further provides the panorama mosaic. Zomet *et al.* [23] introduced a different way of producing mosaics called *cross-slit projection*, which offers the benefit that the generated mosaics are closer to perspective images than traditional pushbroom mosaics. Recent efforts led to additional improvements in the smoothness of strip connections. Wexler and Simakov [20] minimized appearance disagreement error between slices by searching for the best path in the space-time domain. Agarwala *et al.* [1] computed a panorama for a long street scene using Markov Random Field Optimization.

Although the various manifold mosaicing techniques are far less constrained by camera motion model and depth variance than the more traditional image mosaicing algorithms, they do require a dense sampling, such as a video sequence of a static scene taken by one moving camera. In practice, however, obtaining this sampling density is often impractical or prohibitively expensive for typical applications, for example the videoconferencing system introduced in Section1.

[2] As the reference implementation is freely available, Autostitch also serves as a useful basis for comparison with other algorithms.

3 System Details

Our approach is intended to produce smooth mosaicing results even in the case of a highly limited number of input sources and a scene with large depth variance. In summary, starting from calibrated camera inputs, we synthesize a set of virtual images taken from intermediate positions to compensate for the limited sampling rate. It is not necessary to provide rectified inputs to the system, because both of the view synthesis and manifold mosaicing techniques that we employ are robust to the motion model of cameras. We then search in the space-time domain for the best strip stitching path, starting from the first column on the leftmost image to the last column on the rightmost image. The remainder of this section describes each of these steps in further detail.

3.1 Preprocessing

An initial calibration step is required to estimate the intrinsic and extrinsic parameters of the cameras. For this purpose, we use Zhang's [22] method as implemented by Bouguet's calibration toolbox [2]. When applying our algorithm to external data sources, we recover the camera calibration parameters though the structure-from-motion algorithm [10]. Since our system is intended to operate in conjunction with identical high-definition cameras in an environment with controlled illumination, we ignore at present the correction of lens distortion, removal of vignetting, and compensation for exposure or color differences between input images. For lower quality imaging hardware, these factors could be addressed by a number of methods known in the literature.

3.2 Synthesis of Virtual Images

We compensate for a low sampling rate by synthesizing frames that would be generated by a number of virtual cameras positioned between the physical cameras. This decreases the effective baseline distance between neighboring cameras, whether real or virtual, and in turn, minimizes disparity effects between adjacent images, resulting in substantially improved mosaicing results.

The synthesis of virtual frames is performed using the plane sweep algorithm [8], which combines depth from stereo and 3D warping. Plane sweep projects input images onto the image plane of a virtual camera and then sweeps the image plane along the depth axis, examining the color consistency of projected input pixels. Moreover, the algorithm can benefit from implementation on a programmable GPU to provide very fast processing [21] thus likely permitting real-time operation of our image mosaicing system on high-definition video.

The number of virtual camera images is determined as a function of the difference between minimum and maximum disparities observed in the input images and the size of the search window used in the space-time domain. As the difference in disparities increases or the search window shrinks, a higher sampling rate, and thus, a greater number of virtual images, is required to avoid aliasing in the mosaic result. Although beyond the scope of this paper, a more detailed explanation of this topic is provided elsewhere [5][12].

Critical to the stitching operation, the movement (both translation and rotation) between neighboring cameras must be smooth and continuous. Therefore, the virtual sources

are positioned at equal intervals between the physical cameras and their orientations are interpolated smoothly between the rotation matrices of the nearest cameras.

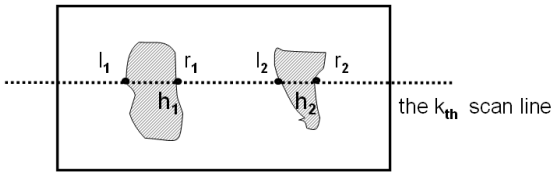


Figure 2: There are two shaded hole regions in this figure. On the k_{th} scan line, points l_1, r_1 and l_2, r_2 are the pairs of non hole region depth value on both sides of the segments across hole region h_1 and h_2 respectively. We fill segment on kth scan line in the hole region h_1 with the bigger depth value of l_1 and r_1 and apply the same way to fill the segment in hole h_2.

When two input cameras are located along a wide baseline, as is the case in our test data, there are likely to be some portions of the 3D scene that are visible by one camera but occluded in the other. This results in holes appearing in the virtual images, which we fill using depth information from neighboring pixels. We scan the hole region row by row. As illustrated in Figure 2, for any row, each hole h_i can be delineated on both sides by the nearest non-hole points, l_i and r_i, with corresponding depths determined by the plane sweep operation. Assuming that the hole exists because of an occluder in one camera view, we estimate its depth by choosing between the two points l_i and r_i the one furthest from the camera. Once all the depth values for the hole regions are estimated in this manner, we can use a foreword forward mapping to project input images onto the virtual image plane at the specified depth, thereby filling in the missing texture of the virtual images. Although more advanced hole-filling algorithms, such as inpainting, are well known, our experiments suggest that the simple strategy described above works sufficiently well for the small holes typically observed in the virtual images.

3.3 Optimal Slice Stitching

Given that no ground truth is available for the image mosaicing result, determining the quality of a particular image mosaic is largely a subjective matter, based on criteria of appearance smoothness and continuity. Wexler *et al.* [20] proposed a metric for these characteristics, and suggested an approach to optimal stitching together of slices to generate a perceptually satisfying mosaic, in which every pair of local neighboring slices should resemble some regions in input frames.

The set of real and virtual camera images are stacked together into a cube, as shown in Figure 3*a*. Each image occupies a slice in $x-y$ space, and these are arranged along the third axis, t.[3]According to the work of adaptive manifold by Peleg *et al.* [14], in the case of translational dominant camera motion model, it is reasonable to choose vertical strips

[3]In video sequences, the meaning of the temporal dimension is obvious; in our case where the images are captured (approximately) simultaneously from a number of cameras, this can be thought of as equivalent to capturing a video sequence of a scene from a single camera moving over time across different input camera positions. Thus, for consistency of terminology, we retain t as a virtual time axis.

from input images to build a mosaicing result. The mosaic is then generated by combining vertical strips from these images along a minimum-cost path in $x-y-t$ space, starting from the first column strip of the first image (at t_0) and ending with the last column strip of the final image (at t_{end}). The cost is determined simply as the sum of squared differences between every pair of adjacent strips being combined. Starting from the initial node in the path, a dynamic programming search is conducted to find the best choice of successive node among all the possible choices within the three-dimensional search volume. This process continues until the final node is reached. An example is illustrated in Figure 3*b*.

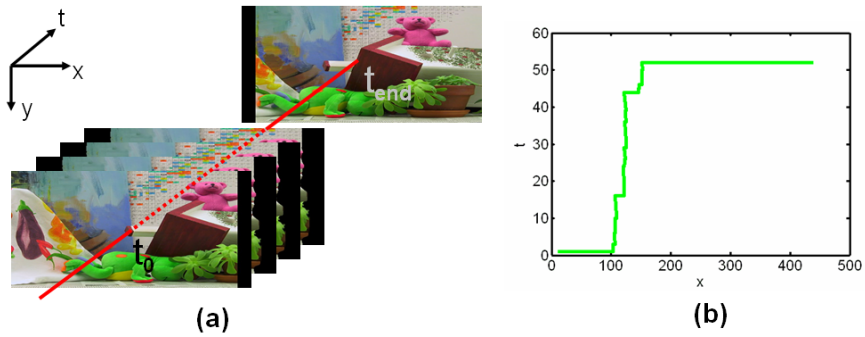


Figure 3: (a)the image cube in the space-time domain and (b)one example of the minimum-cost path found in this domain to stitch the strips from multiple images.

4 Experiments

To evaluate the quality of our algorithm, we compare its results to those of Autostitch [3], which generates excellent mosaics from source images constrained by the conditions outlined in Section 2. This comparison highlights the capacity of our approach to generate significantly improved mosaics from source images exhibiting non-trivial parallax, even with a very low sampling rate.

The first two data sets used for our comparisons are of indoor scenes. The Middlebury teddy data set [7] contains nine color images taken from distinct positions on the same baseline, from which we choose the leftmost and rightmost images as the inputs to our system. Given the range of object depths, the maximum difference in x directional disparities between foreground and background objects is 88 pixels. The second example, collected by Seitz [17] exhibits a maximal disparity difference of 126 pixels while the outdoor (house) scene exhibits a lower disparity difference of only 46 pixels.

Autostitch first deforms the two input images, compressing objects closer to the cameras and expanding distant objects to equalize their respective disparities. This normalizes the amount of overlap between the images, allowing for a smooth combination of the two deformed inputs.

The results of Autostitch, seen in Figures 4 and 5, contain a foreground region at the bottom of the mosaic, which has shrunk relative to the background at the top. We verified that this differential stretching effect is due to image content by flipping the source images

upside down. As expected, this produces a mosaic in which the foreground, now at the top of the image, shrinks relative to the background at the bottom. In contrast, the results of our algorithm are free of such deformation, as the parallax effects have been greatly reduced by the incorporation of synthesized virtual images. Furthermore, our mosaics do not exhibit the unpleasant ghosting effects seen in the Autostitch results, stemming from the difficulty of accurate alignment given the huge depth variances in the scene.

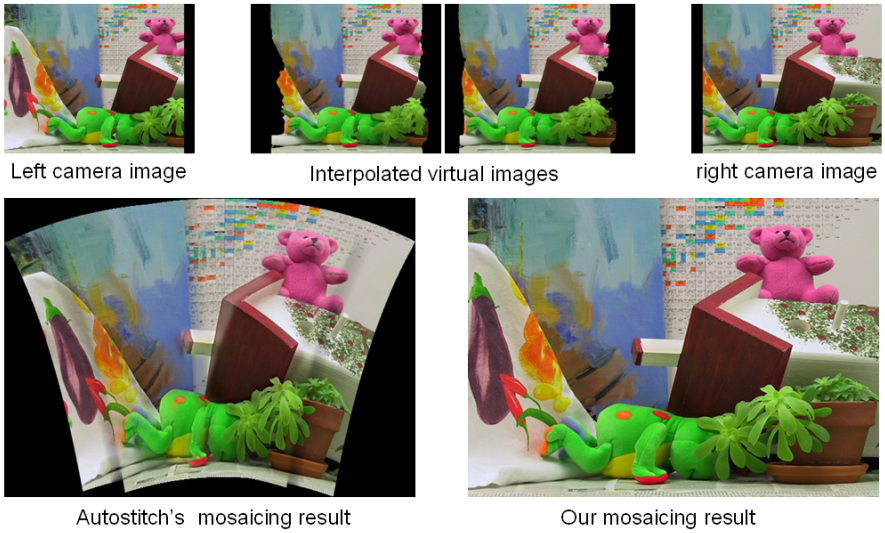


Figure 4: Experimental results from the Middlebury teddy data set.

The third data set used for our experiments is taken from an outdoor scene of a house with trees in the background and right foreground. This example has disparity difference within the range that Autostitch can handle. The resulting mosaic, pictured in Figure 6, still exhibits some radial distortion as well as ghost errors around the window, while our result, although free of radial distortion, contains ghost errors on the garage door. Potentially contributing factors to this error include the accuracy of camera calibration, the large textureless regions in the input images, and robustness of the plane sweep algorithm to color inconsistency.

In the experimental results, above, our algorithm used only the color channel of the virtual images, searching for the best stitching path in the spatiotemporal volume according to the color consistency principle. We also tested the algorithm using the depth channel of virtual images, connecting stripes only when their contents are similar in depth.

As shown in Figure 7, searching by depth consistency finds a similar path as with the color consistency rule. The results are slightly noisier than those of the earlier mosaicing example, based on color consistency, seen in Figure 4. This is likely because the estimation of depth by the plane sweep operator is less robust than that of color. However, the test confirms that depth cues are also important to determining alignments between input images. This may prove significant when generalizing our algorithm to dynamic scenes, with objects moving about the environment. Otherwise, we are required to recompute the entire stitching path for each frame, which generally results in observable jitter.

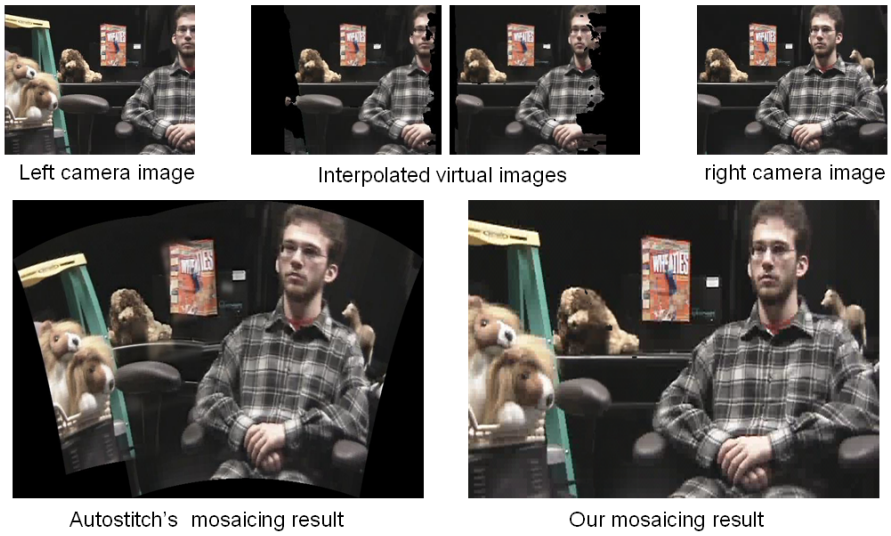


Figure 5: Experimental results from an indoor scene with greater depth variance.

5 Conclusions and Future work

Issues of parallax and camera motion rate prevent traditional image mosaicing algorithms from generating perceptually acceptable panoramas in many situations. The approach we introduce combines previous knowledge of image-based rendering and manifold mosaicing to overcome some of these limitations and produce reasonable mosaics from sparse input sources, situated along a wide baseline.

Our system may be improved along a number of dimensions. Taking advantage of the power of the programmable GPU could greatly accelerate the processing speed and allow for real-time operation on live video. The quality of synthesized virtual images could be improved by a global optimization algorithm and by tracking the movements of people (or other dynamic objects) in the scene.

References

[1] A. Agarwala, M. Agrawala, M. Cohen, D. Salesin, and R. Szeliski. Photographing long scenes with multi-viewpoint panoramas. *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 853–861, 2006.

[2] J.V. Bouguet. Camera calibration toolbox for matlab. `http://www.vision.caltech.edu/bouguetj/calib_doc/`. 2003.

[3] M. Brown and D.G. Lowe. Autostitch :: a new dimension in automatic image stitching. `http://www.cs.ubc.ca/~mbrown/autostitch/autostitch.html`.

[4] M. Brown and D.G. Lowe. Recognising panoramas. *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 1218, 2003.

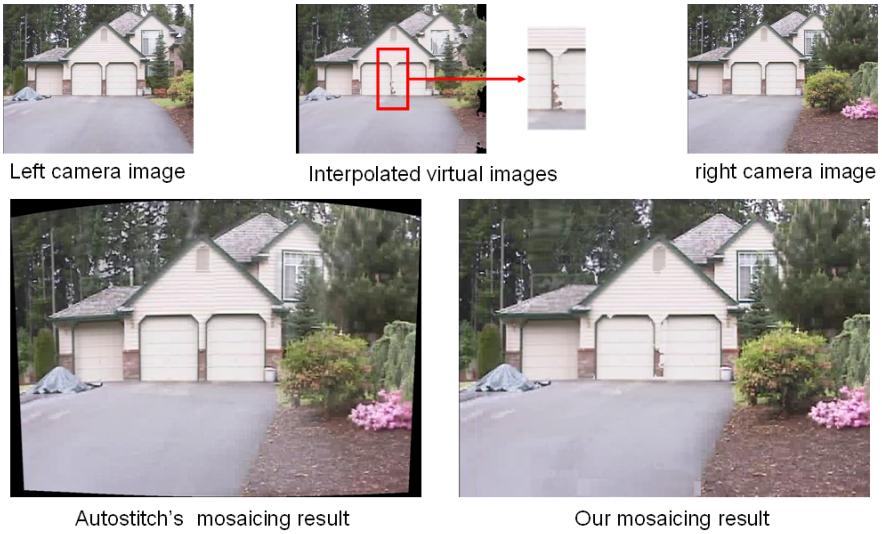


Figure 6: Experimental results from an outdoor scene with small depth variance. The boxed region of the synthesized virtual image is enlarged to the right in order to illustrate the source of the ghost error around the garage door.

[5] J.X. Chai, S.C. Chan, H.Y. Shum, and X. Tong. Plenoptic sampling. *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 307–318, 2000.

[6] S.C. Chen. Quicktime vr: an image-based approach to virtual environment navigation. *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 29–38, 1995.

[7] Middlebury College. Stereo vision research page. `http://cat.middlebury.edu/stereo/newdata.html`.

[8] R. Collins. A space-sweep approach to true multi-image matching. *IEEE Computer Vision and Pattern Recognition*, pages 358–363, 1996.

[9] R. Gupta and R.I. Hartley. Linear pushbroom cameras. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(9):963–975, 1997.

[10] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, New York, NY, USA, 2000.

[11] A. Levin, A. Zomet, S. Peleg, and Y. Weiss. Seamless image stitching in the gradient domain. *Hebrew university technical report:2003-82*, 2003.

[12] Z.C. Lin and H.Y. Shum. A geometric analysis of light field rendering. *Int. J. Comput. Vision*, 58(2):121–138, 2004.

[13] S. Peleg and J. Herman. Panoramic mosaics by manifold projection. *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, page 338, 1997.

[14] S. Peleg, B. Rousso, A. Rav-Acha, and A. Zomet. Mosaicing on adaptive manifolds. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(10):1144–1154, 2000.

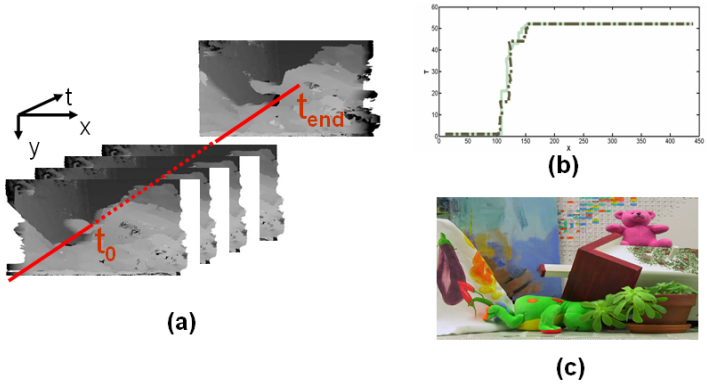


Figure 7: (a) The group of real and virtual images in the spatiotemporal domain, (b) the minimum-cost path found by searching for consistency in the depth channel (light solid line) and in the color channel (dark dashed line), (c) the corresponding color channel mosaicing result based on the path of depth channel (light solid line) in (b).

[15] A. Rav-Acha, Y. Shor, and S. Peleg. Mosaicing with parallax using time warping. *CVPRW '04: Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04) Volume 11*, page 164, 2004.

[16] A. Román and H.P.A. Lensch. Automatic multiperspective images. *Rendering Techniques 2006: Eurographics Symposium on Rendering*, pages 161–171, 2006.

[17] S.M. Seitz and J. Kim. The space of all stereo images. *Int. J. Comput. Vision*, 48(1):21–38, 2002.

[18] H.Y. Shum and He L.W. Rendering with concentric mosaics. *Computer Graphics in SIGGRAPH '99*, pages 299–306, 1999.

[19] H.Y. Shum and R. Szeliski. Construction and refinement of panoramic mosaics with global and local alignment. *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*, page 953, 1998.

[20] Y. Wexler and D. Simakov. Space-time scene manifolds. *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, pages 858–863, 2005.

[21] R. Yang, M. Pollefeys, H. Yang, and G. Welch. A unified approach to real-time multiresolution. *International Journal of Image and Graphics, 2004*, 2004.

[22] Z.Y. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.

[23] A. Zomet, D. Feldman, S. Peleg, and D. Weinshall. Mosaicing new views: The crossed-slits projection. *IEEE PAMI*, pages 741–754, 2003.

Bayesian Surface Estimation from Multiple Cameras Using a Prior Based on the Visual Hull and its Application to Image Based Rendering

Adam Bowen, Andrew Mullins, Roland Wilson and Nasir Rajpoot
Department of Computer Science
University of Warwick, UK
{fade,andy,rgw,nasir}@dcs.warwick.ac.uk

Abstract

The problem of visible surface estimation for image-based rendering is tackled using a new approach, which combines visual hull and surface estimation techniques. It is shown that the new method combines the best features of both approaches, being more robust than direct surface element estimation and more flexible than the visual hull. The new method uses an estimate of the visual hull as a prior on the ill-posed problem of surface element estimation. To improve the computational preformance of the algorithm, a multiresolution approach to surface patch estimation is used. The patches thus estimated can then be tracked over time, to provide an accurate model of the surface geometry, which is then used for estimation of the scene from arbitrary viewpoints. After a brief description of the algorithm, results are presented to show the improvement in performance which can be obtained using either technique alone. The paper concludes with a discussion of extensions to the work currently under investigation.

1 Introduction

The problem of image-based rendering has been much studied and ranges in complexity from simple interpolation techniques [2], [3], [8] to geometry estimation and space carving [4], [19], [16], [20], [5]. While simple interpolation approaches are adequate if a sufficiently dense set of images of the scene is available, in many practical situations, the number of cameras limits the accuracy of interpolations and leads to visible artefacts in the reconstructions. The alternative is an explicit computation of surface geometry, which allows more conventional, graphics based rendering to be used. Unfortunately, even with multiple cameras, estimation of visible surfaces remains an ill-posed problem, especially if the objects imaged are as complex as the human body. Two common approaches have been used to estimate scene geometry: surface estimation techniques [4], [19], [15], [18], and the so-called visual hull [7], [1], [16], [20]. Although the former is inherently more powerful, it suffers from the ill-posed nature of the problem: it depends heavily on the surface texture and smoothness of the 3-D shape of the objects in the scene; it is also highly demanding computationally. Visual hull techniques, on the other hand, require accurate segmentation of the projected images

and make simplifying assumptions about the convexity of the object [10], require a large amount of memory, and suffer from quantization effects [9], but they lead to robust estimates; they are also reasonably fast computationally. A method which combines the features of the two seems like an obvious way to overcome the limitations of either.

Previous work in this area has used both determinstic [19] and simple stochastic methods [6] for hull refinement. In this paper, we present such a method, which uses a stochastic Bayesian framework for the computation of planar surface elements but employs a prior on those elements, derived from the visual hull. Such surface elements are a common respresentation of surface geometry [15], [18], [17], and provide a more general representation than similar surface descriptions such as wireframe meshes. The input to the algorithms are a set of video sequences together with camera calibration information for those sequences. The calibration information is obtained separately using the methods described in [22].

After a brief discussion of the surface representation and a description of the estimation algorithms, we present some results to show the performance of the new method, which significantly outperforms our earlier methods, giving more accurate surface estimates in a reduced computation time. The paper is concluded with a discussion of the power and limitations of the new approach and suggestions for further work.

2 Surface Estimation Method

First, a multiresolution segmentation algorithm is applied to the images captured from the scene to separate figure and background. From the silhouettes derived from the segmentation, a view-dependent visual hull is constructed following a similar method to that described in [11]. From the visual hull the depth and surface orientation along any ray in the scene may be derived using total least squares, a robust method of fitting planes to surface data [12]. These estimates are used as a prior in a multiscale particle filter, which provides the final surface estimates, in terms of a number of disjoint quadrilaterals, corresponding to square blocks in each image of the scene. The patches can then be tracked over time using a second particle filter [14], and used to reconstruct arbitrary views of the scene using an adaptation of a conventional graphics renderer, giving real-time reconstructions [13], [17].

2.1 Surface Representation

A view dependent surface model is defined as follows: partition each input image into pixel blocks and assume each block is the projection of a quadrilateral region of a surface in world space. For notational convenience it is assumed that every block has a unique index n. The class of quadrilaterals corresponding to image blocks will subsequently be refered to as 'patches'.

The centroid of a patch which corresponds to a block n, must lie along the line which passes through both the imaging camera, and the focal plane at the point which corresponds to the centre of the block (such lines will be refered to as 'pixel rays' for obvious reasons). A compact representation of the position of a patch is its distance d_n from the camera, along this pixel ray.

The orientation of a patch must lie between face-on, and almost perpendicular to, the imaging camera. In an appropriately chosen Cartesian frame where the z-axis aligns with the pixel ray for the centre of block n, the orientation of the patch may be represented using the first two components (i_n, j_n) of its surface normal, with the third component having unit length. This representation allows all allowable orientations, yet prevents the patch from becoming oriented perpendicular, or beyond perpendicular, to the imaging camera. The complete representation of a patch corresponding to block n is then

$$x_n = (d_n, i_n, j_n). \tag{1}$$

For convenience in the later discussion, $f_n(x_n)$ will denote the plane defined by x_n in some fixed Cartesian frame common to all patches.

2.2 Locally Adaptive Foreground-Background Segmentation

For a camera c in a data set, its image will be denoted I_c, and its Gaussian pyramid decomposition $I_c^{[0]} \ldots I_c^{[M]}$, where $I_c^{[0]} = I_c$. For a pixel (u,v) in an image level $I_c^{[m]}$, its labelling $s(u,v)$ as either foreground or background is performed using a likelihood model derived from a segmentation of the previous image pyramid level $I_c^{[m+1]}$. One global approach using the above idea would be to evaluate the likelihood of a labelling $s(u,v) = l$ as

$$p_s(I_c^{[m+1]} | s(u,v) = l) = \frac{1}{K} \sum_{k=0}^{K} N(I_c^{[m]}(u,v)\,; \mu_l^{[k]}, \sigma_l^{[k]}), \tag{2}$$

with the K Gaussian mixture components being derived from the colours of the pixels in class l at the previous image level. However, unless the number of components is very large, image details will not feature prominently in the mixture and may be misclassified.

Adopting a more local approach, our algorithm proceeds as follows: firstly each pixel is labelled using the classification of its 'parent' pixel from the previous image level, in quad-tree fashion. Then the image is partitioned into blocks and the segmentation refined with a two pass approach:

Pass one: For each block: if every pixel within the block and that of its neighbours is of the same class, mark the block as classified.

Pass two: The remaining unclassified blocks form a 'corridor of uncertainty' [21]. For each unclassified block,

i Search within an increasing radius until one or more blocks are found from each class which were classified in the first pass.

ii Form a Gaussian mixture colour likelihood model for both classes by clustering the pixel values within these classified blocks.

iii Classify each pixel within the current block as either foreground or background based on the local likelihood model.

As part of the third step, it is possible to use a Markov random field using the likelihood models and a smoothness prior as in [23], although this was found to not improve the results significantly given the extra computation required. The only remaining issue is how a classification is produced at the lowest resolution. It was found that a simple foreground-background model based on a background reference image provided a reasonable starting point.

2.3 Surface Prior Construction

Initially, a view dependent hull is constructed. For every pixel u,v in a particular image level $I_c^{[m]}$, it is possible to estimate a point which lies on the surface of the visual hull as follows:

i Generate a foreground-background segmentation for every camera.

ii Intersect the pixel ray for each pixel u,v labelled as foreground in image $I_c^{[m]}$ with the back-projected foreground regions in all other cameras $c' \neq c$ to produce zero or more lines in world space.

iii The closest line end point p_{uv} to camera c is the point on the visual hull which projects into pixel u,v.

The algorithm to construct a surface prior $\hat{x_n}$ for block n then proceeds as follows:

i Calculate $M = \frac{1}{|P_n|} \sum_{(u,v)\in P_n} (p_{uv} - \bar{p})(p_{uv} - \bar{p})^T$, where $\bar{p} = \frac{1}{|P_n|} \sum_{(u,v)\in P_n} p_{uv}$, and where P_n is the set of pixels in block n.

ii Compute the eigenvectors and eigenvalues of M.

iii The eigenvector corresponding to the smallest eigenvalue is the normal to the best-fit plane through the data, and together with $\bar{p}$ defines a plane y_n.

The prior can then be chosen for block n and has mean $\hat{x_n} = f_n^{-1}(y_n)$, and covariance $\hat{\sigma}_n$ which is chosen empirically.

2.4 Multiresolution Particle Filter

A particle filter is used which works across scale to produce surface estimates. For each patch at a given image level, the algorithm draws a set of samples $X = \{x^{[1]} \ldots x^{[S]}\}$ from an importance sampling function,

$$X \sim q(x_n | x_{h(n)}, I^{[m]} \ldots I^{[M]}), \tag{3}$$

where $h(n)$ is the parent block of block n.

The samples drawn from the importance sampling distribution are weighted according to

$$w_n^{[s]} \propto w_{h(n)}^{[s]} \frac{p(I^{[m]} | x_n^{[s]}) p(x_n^{[s]} | x_{h(n)}^{[s]})}{q(x_n | x_{h(n)}, I^{[m]} \ldots I^{[M]})}, \tag{4}$$

where the process model for the parameters of block n is simply

$$p(x_n | x_{h(n)}) = N(x_n;\ f_n^{-1}(f_{h(n)}(x_{h(n)})), \sigma_p) \tag{5}$$

i.e. the parameters corresponding to the same plane as that defined by its parent, but with some process noise with covariance σ_p added.

The measurement likelihood $p(I^{[m]} | x_n)$ is defined as follows: for any pixel (u,v) in block n, intersecting its pixel ray with the plane $f_n(x_n)$ produces a point in world space.

This point may be projected into any other camera's images and the colour of the original pixel compared to that in the projected location. Intuitively, the more similar the colours, the more likely that x_n are the correct parameters for the block. This mapping between a pixel (u,v) belonging to a block n in camera c, and the pixel (u',v') to which it projects in camera c' given that the block has parameters x_n is defined as $(u',v') = g^{c\rightarrow c'}(u,v,x_n)$.

Following this intuition we define a likelihood for a patch, given a set of multiscale image data from the cameras. The likelihood of x_n, given one camera's image level $I_{c'}^{[m]}$ is

$$p(I_{c'}^{[m]}|x_n) = \frac{1}{|P_n|}\sum_{(u,v)\in P_n} N(I_{c'}^{[m]}(g^{c\rightarrow c'}(u,v,x_n)) ; I_c^{[m]}(u,v), \sigma), \tag{6}$$

That is, the colour of the re-projected pixel is assumed to be normally distributed, with covariance σ about the colour of the pixel from the original block. Given all the images $I^{[m]} = \{I_1^{[m]} \ldots I_C^{[m]}\}$ from all cameras in the data set at level m, the likelihood of a patch is

$$p(I^{[m]}|x_n) = \frac{1}{|N_c|}\sum_{c'\in N_c} p(I_{c'}^{[m]}|x_n), \tag{7}$$

where N_c are the spatial neighbours of camera c.

3 Results

Our locally adaptive segmentation provides a clear advantage over the Markov random field approach in dealing with background regions of images onto which a shadow has been cast. Figure 4 highlights the class boundary for the two methods for an image taken from one of our data sets. Although the Markov random field has corrected the white regions on the t-shirt which are classified as background under using a naive approach, it has also misclassified several areas of shadow close to the body as foreground. The locally adaptive model as prevented this misclassification.

The results in table 3 indicate the mean squared error in colour across all patches when they are reprojected into neighbouring cameras, using different patch estimation algorithms for two data sets. The first contains a static person, whilst the second contains a person in motion which presents a significant challenge for the likelihood model due to motion blur. 'MSMC' is the multi-scale Monte-carlo algorithm which simply estimates patches using a particle filter across scale, but a weak prior. The 'VH' algorithm produces surface estimates derived from the visual hull alone. The 'VH + MC' algorithm estimates the patches using Monte-carlo simulation at the finest resolution using the visual hull to provide a prior. Finally, the 'VH + MSMC' algorithm is the full multiscale Monte-carlo algorithm using the visual hull as a prior, as described in section 2.4. This algorithm clearly outperforms all other approaches.

The ultimate objective of this work is to render images from a new point of view not in the original data set. Figure 2 shows three reconstructions produced from the patches using the 'MSMC' algorithm, the 'VH' algorithm and the 'VH + MSMC' algorithm.

Data set	MSMC	VH	VH + MC	VH + MSMC
1	1571.9	2563.4	2318.5	1374.3
2	n/a	9179.8	9268.7	5585.8

Table 1: MSE of estimated patch projections.

4 Conclusions and Further Work

We have shown the power of a combining hull and surface estimation within a Bayesian framework. It significantly increases the accuracy and speed of object geometry estimation. In addition, we have proposed a novel multiscale method for producing the foreground-background segmentations which are used to generate the visual hull. As a part of future work we would like to investigate the possibility of using hull information for the estimation of frame-to-frame motion.

References

[1] Adrian Broadhurst, Tom Drummond, and Roberto Cipolla. A probabilistic framework for space carving. pages 388–393.

[2] E. Chen and L. Williams. View interpolation for image synthesis, 1993.

[3] S. Chen. Quicktime vr — an image-based approach to virtual environment navigation, 1995.

[4] S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen. The lumigraph, 1996.

[5] A. Hilton and J. Starck. Multiple view reconstruction of people.

[6] J. Isidoro and S. Sclaroff. Stochastic refinement of the visual hull to satisfy photometric and silhouette consistency constraints.

[7] Kiriakos N. Kutulakos and Steven M. Seitz. A theory of shape by space carving. Technical Report TR692, 1998.

[8] Marc Levoy and Pat Hanrahan. Light field rendering. *Computer Graphics*, 30(Annual Conference Series):31–42, 1996.

[9] M. Li, M. Magnor, and H. Seidel. Improved hardware-accelerated visual hull rendering, 2003.

[10] Ming Li, Hartmut Schirmacher, Marcus Magnor, and Hans-Peter Seidel. Combining stereo and visual hull information for on-line reconstruction and rendering of dynamic scenes.

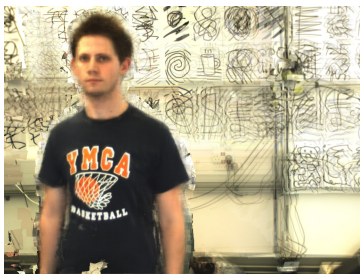
[11] Wojciech Matusik. Image-based visual hulls. In *Master of Science Thesis*, 2001.

[12] Niloy J. Mitra and An Nguyen. Estimating surface normals in noisy point cloud data. In *Proceedings of the nineteenth annual symposium on Computational geometry*, pages 322–328, 2003.

(a) Fixed foreground-background model. (b) Adaptive foreground-background model.

Figure 1: An input image with the border between foreground and background image overlayed for two different algorithms.

(a) Reconstructed from patches estimated with 'MSMC'. (b) Reconstructed from patches estimated with 'VH'.



(c) Reconstructed from patches estimated with 'VH + MSMC'.

Figure 2: Reconstructions obtained after using various patch estimation algorithms.

[13] Andrew Mullins, Adam Bowen, Roland Wilson, and Nasir Rajpoot. Estimating planar patches for light field reconstruction. In *The Proceedings of the British Machine Vision Conference*, 2005.

[14] Andrew Mullins, Adam Bowen, Roland Wilson, and Nasir Rajpoot. Surace estimation and tracking using sequential mcmc methods for video based rendering. In *To be published in: The Proceedings of the International Conference on Image Processing ICIP07*, 2007.

[15] Don Murray and James J. Little. Patchlets: Representing stereo vision data with surface elements. In *Proceedings of the Seventh IEEE Workshops on Application of Computer Vision.*, volume 1, pages 192–199, 2005.

[16] P J Narayanan and Takeo Kanade. Virtual worlds using computer vision. In *Proceedings of the 1998 IEEE and ATR Workshop on Computer Vision for Virtual Reality Based Human Communications*, pages 2 – 13, January 1998.

[17] Hanspeter Pfister, Matthias Zwicker, Jeroen van Baar, and Markus Gross. Surfels: Surface elements as rendering primitives. In Kurt Akeley, editor, *Siggraph 2000, Computer Graphics Proceedings*, pages 335–342. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.

[18] Tobias Pietzsch and Axel Gromann. A method of estimating oriented surface elements from stereo images. In *Proceedings of the British Machine Vision Conference*, volume 1, 2005.

[19] P. Ramanathan, E. Steinbach, P. Eisert, and B. Girod. Geometry refinement for light field compression.

[20] S. Seitz and C. Dyer. Photorealistic scene reconstruction by voxel coloring, 1997.

[21] Michael Spann and Roland Wilson. A quad-tree approach to image segmentation which combines statistical and spatial information. In *Pattern Recognition 18*, volume (3-4), pages 257–269, 1985.

[22] Tomáš Svoboda, Daniel Martinec, and Tomáš Pajdla. A convenient multi-camera self-calibration for virtual environments. *PRESENCE: Teleoperators and Virtual Environments*, 14(4):407–422, August 2005.

[23] Yang Wang, Kia-Fock Loe, Tele Tan, and Jian-Kang Wu. A dynamic hidden markov random field model for foreground and shadow segmentation. In *Proceedings of the Seventh IEEE Workshops on Application of Computer Vision*, volume 1, pages 474–480, 2005.

This work was supported by the UK EPSRC.

A multi-class SVM classifier ensemble for automatic hand washing quality assessment

D.F. Llorca, F. Vilarino, Z. Zhou and G. Lacey
Graphics, Vision and Visualisation Group (GV2)
Computer Science Dep. Trinity College Dublin, Rep. of Ireland
llorca@depeca.uah.es

Abstract

Hand washing is a critical activity in preventing the spread of infection in health-care environments. Several guidelines recommended a hand washing protocol consisting of six steps that ensure that all areas of the hands are thoroughly cleaned. In this paper, we describe a novel approach that uses a computer vision system to monitor the user's hands motions in order to ensure that the hand washing guidelines are followed. This work presents two main contributions: a description of a system which delivers robust segmentation of the hands using a combination of colour and motion analysis, and the implementation of a multi-class classification of the hand gestures using a SVM ensemble. The system performance is analysed and compared with human performance, showing an accuracy close to that of human experts.

1 Introduction

Hand washing is widely acknowledged to be the single most important activity for reducing the spread of infection in clinical environments. It is extremely important that health care professionals use the correct technique. This means that no areas of the hands should be missed. According to several guidelines [15] the correct hand washing procedure should comprise six different poses as depicted in Figure 1:

Pose 1 Pose 2 Pose 3 Pose 4 Pose 5 Pose 6

Figure 1: *Pose 1*: Rub palm to palm. *Pose 2*: Rub palm over the back of the other one. *Pose 3*: One hand over back of the other hand and rub fingers. *Pose 4*: Rub palm to palm with fingers interlaced. *Pose 5*: Wash thumbs of each hand separately using rotating movement. *Pose 6*: Rub finger tips against the opposite palm using circular motion.

In the last few years, several authors have addressed the problem of single hand gesture recognition [16], and recently gesture recognition has been applied to hand washing [7]. Among the wide range of vision-based hand gesture recognition techniques in the literature, we emphasize: Histograms of oriented gradients (HOG) have been used as feature

vectors in hand gesture classification [5]. Spatio-temporal hand gesture recognition using neural networks has been introduced in [19] and [12], and recognition of gestures using hidden Markov models (HMM) has been also widely studied [16],[18], [14]. In [10], a robust hand posture detection method based on the Viola and Jones detector [20] is proposed; in addition, a frequency analysis-based method for class separability estimation is used and a comparison of non-cascaded and cascaded detectors is presented.

In this work, a novel system for hand washing quality assessment using computer vision is proposed. Hands/arms segmentation is achieved by combining skin and motion features to ensure a robust region of interest (ROI) selection process which is independent of lighting conditions or reflectivity of the sink (ceramic, stainless steel, etc.). Distributed HOG are used to create the feature vector which will be dispatched to the classifier. In order to recognize the six different poses of the correct hand washing technique, a support vector machine (SVM) [1] with a one-against-one multi-class approach is used.

This paper is organised as follows: Section 2 provides a detailed description of the ROI selection process. The feature extraction method is described in Section 3, and the description of the classifier used to recognise the different poses is shown in Section 4. The results achieved up to date and their comparison with human performance are presented in Section 5. Finally, our conclusions and the description of our future lines of research are presented in Section 6.

2 ROI Selection

To increase the reliability of the classification it is necessary to identify the hands in the images and to locate a region of interest around the hands. This is important for both the case when the hands are joined and when they are separate. This step is crucial, since both the training and performance of the classifier depend on the manner in which the ROI is calculated.

Typically in the case of an acquisition process obtained from a stationary camera, background subtraction would be a good segmentation strategy. However, in this application the foreground drastically changes the background, not only its intensity but also its colour. With fixed lighting conditions in stainless steel sinks, for example, the hands/arms may tend to be confused with background due to the multiple reflections and changes in colour on the steel surface. In addition, adaptive background subtraction approaches appear not to improve the results because almost all the time the hands are moving around the same area, and consequently the hands are gradually integrated with the background. Accordingly, other techniques should be used. Hands/arms segmentation is thus posed in this work as a problem which integrates both skin detection and motion analysis.

2.1 Skin Colour Detection

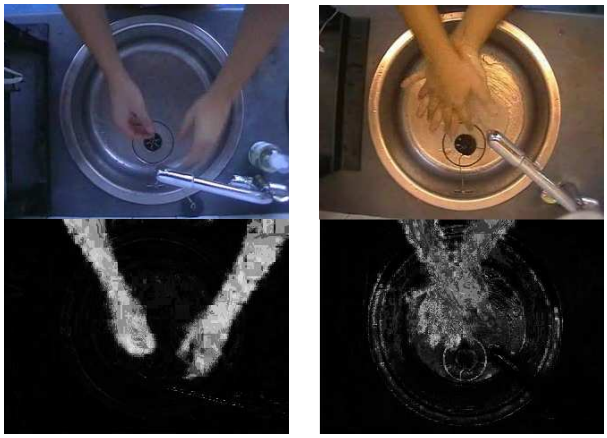
Skin detection plays an important role in several applications such as face detection, searching and filtering image content on the web, video segmentation, face/head tracking, etc. In our case a non-parametric skin modeling with Bayesian models [9] is applied, in which the skin and non-skin colours are modelled through histograms. We quantize the colour space *RGB* to a number of bins N^{bins} and count the number of colour pixels in each bin N_{skin} for skin class as well as $N_{non-skin}$ for non-skin class. Finally we normalise each bin to get the discrete conditional *skin/non-skin* colour distribution

$P(rgb|skin)/P(rgb|non\text{-}skin)$ [9]. A skin binary map can be made upon a properly selected threshold over the posterior probabilities obtained.

Although the described scheme can successfully deal with lighting changes, the varying lighting conditions of the wash hand basin requires a lightning compensation step in order to achieve more robust estimates of skin tone. We propose the use of the grey-world approach implemented by Chen et al. [6]. This method is based on the assumption that the spatial average of surface reflectance in a scene is achromatic. Since the light reflected from an achromatic surface is changed equally at all wavelengths, it follows that the spatial average of the light leaving the scene will be the colour of the incident illumination. In addition, even though lighting compensation reduces the variability in skin tones detected, there remains a significant challenge to robust segmentation in real world situations, i.e., reflections due to stainless steel wash hand basins. This is a factor of both the skin tones of the user and the lighting conditions, and an example of this can be observed in Figure 2. In Figure 2 (a) reliable segmentation is achieved, while in Figure 2 (b) additional light from a window produces certain reflections of the hands on the stainless steel. Consequently, in situations where stainless steel wash hand basins are used, additional features are required to achieve robust segmentation. In our approach we propose the use of features based on motion analysis.

2.2 Skin and Motion Segmentation

The idea on which our proposal for robust hand segmentation is based consists of computing an overall image that integrates both skin and motion features. For each skin pixel, which could be a false positive, we calculate the motion vector obtained by means of optical flow analysis [13], after applying an average filter over the motion magnitude image. If the averaged motion magnitude is greater than a fixed threshold T_{ID} the probability estimate is increased. If it is less than T_{ID}, then the probability is decreased. We implement



(a) (b)

Figure 2: Upper row: Original images. Lower row: Skin probability maps. (a) With favorable, and (b) unfavorable lighting conditions.

this increase/decrease effect by means of the coefficients I and D, defined according to the following equations:

$$I = I_F(1 - \frac{1}{\exp(\frac{M_i - T_{ID}}{0.5T_{ID}})}); \qquad D = D_F \frac{1}{\exp(\frac{M_i}{0.5T_{ID}})} \tag{1}$$

where I_F and D_F are the increase and decrease factors, M_i is the motion magnitude of pixel i, and T_{ID} is the threshold value which decides if there is going to be an increase or a decrease in the output image. The increase function should reach the maximum value quickly, whereas decrease function must be less steep. This is because we want to quickly highlight motion as soon as it is detected and penalise its absence more slowly. Thus skin pixels due to metal sinks and specular reflections are slowly removed while the hands/arms are correctly segmented only if they are moving. This architecture does not resolve the issue of hand detection while the hands are not moving. However, in this application the hands rarely stop moving and one can argue that if they do stop no effective hand washing is being performed. Figure 3 shows a summary of the global architecture described so far for the hands detection based on both colour and motion information.

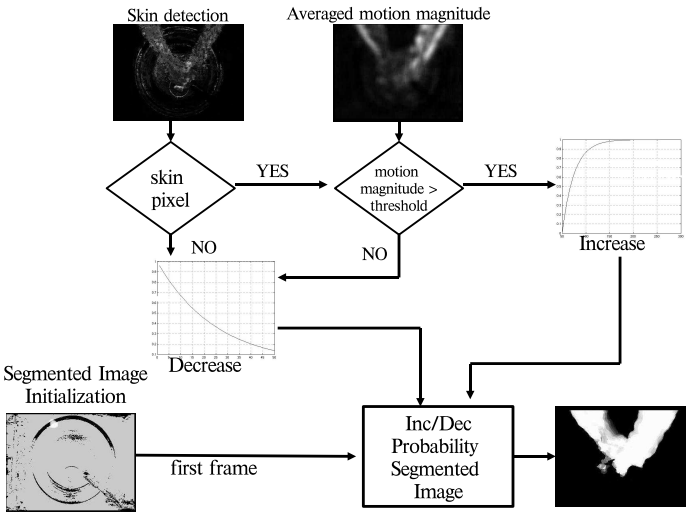


Figure 3: Skin and motion detector for hands/arms segmentation scheme

Following detection, we apply a connected components analysis to identify the hands and arms. Geometric analysis of the shapes leads to the computing of the location of the boundaries of the ROI. Next, a region of interest covering the hands is selected. The vertical symmetry axis is used for computing upper and lower boundaries, which are then used for obtaining the lateral ones. This process is depicted in Figure 4.

3 Feature Extraction

Choosing discriminating and independent features is key to any pattern recognition algorithm being successful in classification. We use local histograms of oriented gradients as

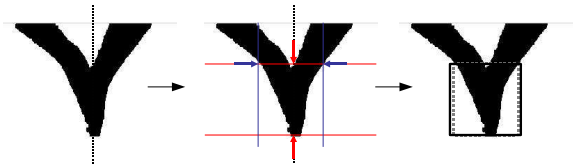


Figure 4: Hands ROI selection: First a vertical symmetry axis is obtained, then the vertical bounds are delimited, and finally the horizontal bounds are obtained.

our single frame feature extraction method, which has reported good performance results in recent applications [4]. The aim of this method is to describe an image by a set of local histograms. These histograms count occurrences of gradient orientation in a local part of the image. The procedure is a follows: First the colour image is converted to grey level, then the gradient is calculated, and next the image is split into cells which are defined as square regions with a predefined size in pixels. For each cell, we compute the histogram of gradients by accumulating votes into bins for each orientation. Votes can be weighted by the magnitude of the gradient vector, so that the histogram takes into account the importance of the gradient at a given point.

Due to the variability in the images, it is necessary to normalise the cell histograms. Cell histograms are locally normalised according to values of the neighbored cell histograms. The normalisation is done among a group of cells, which is referred to as a block. A normalisation factor is then computed over the block and all histograms within this block are normalised according to this normalisation factor. We use the L2-norm scheme:

$$v \rightarrow \frac{v}{\sqrt{\|v\|_2^2 + \varepsilon}} \tag{2}$$

where ε is a small regularization constant needed because sometimes empty gradients are going to be evaluated. Note that according to how each block has been built, a histogram from a given cell can be involved in several block normalisations. Thus we are going to have some redundant information which, according to the work of Dalal et al. [4], improves the performance. Figure 5 shows a graphical description of this method.

When all histograms have been computed for each cell, we build the description vector of an image by concatenating all histograms into a single vector. In order to compute the vector dimension several parameters have to be taken into account: ROI dimension, cell size, block size, number of bins and number of overlapped blocks. Joining together all the elements described so far, the final recount of features is a follows: the ROI size is 128×128, each window is divided into cells of size 16×16 and each group of 2×2 cells is integrated into a block in a sliding fashion, so blocks overlap with each other. Each cell consists of a 16-bin HOG and each block contains a concatenated vector of all its cells. Each block is thus represented by a 64 feature vector which is normalised to an L2 unit length. Thus, in our case we have feature vectors of 3.136 dimensions. The following chart summarizes the list of all the features included in the final feature vector:

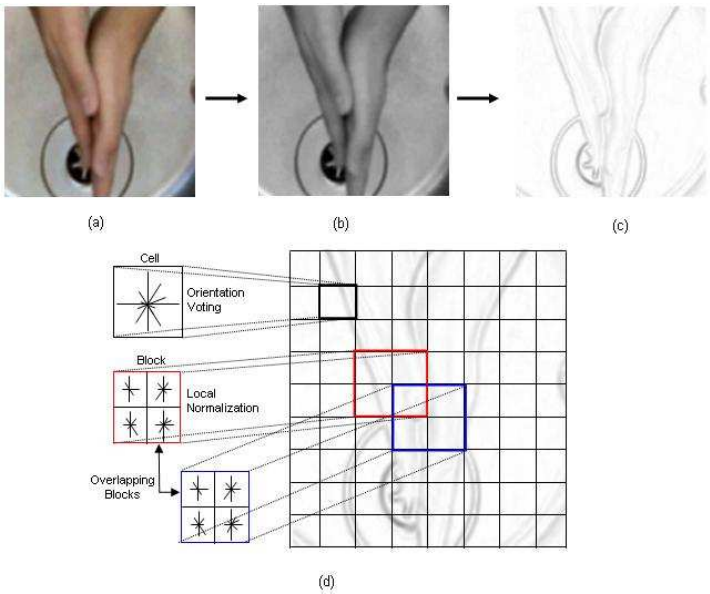


Figure 5: HOG description, (a) original image, (b) grey image, (c) gradient image, (d) HOG, cells distribution and blocks normalisation

$Cells_{Row} = 128/16 = 8$	$Cells_{Col} = 128/16 = 8$
$CellsBlock_{Row} = 32/16 = 2$	$CellsBlock_{Col} = 32/16 = 2$
$OverlapBlocks_{Row} = 8-2+1 = 7$	$OverlapBlocks_{Column} = 8-2+1 = 7$
$\mathbf{VectorDIM = 7*7*2*2*16 = 3136}$	

4 The SVM Classifier Ensemble

The support vector machine classifier is a binary classifier algorithm that looks for an optimal hyperplane as a decision function in a high dimensional space [1]. It is a kind of example-based machine learning method for both classification and regression problems. This technique has several features that make it particularly attractive. Traditional training techniques for classifiers, such as multi-layer perceptrons (MLP) use empirical risk minimisation and only guarantee minimum error over the training set. In contrast, SVM bases on the structural risk minimisation principle [1] which minimises a bound on the generalisation error and therefore should perform better on novel data.

In order to perform the multi-class classification the one-against-one method is proposed, with which $k(k-1)/2$ different binary classifiers are constructed and each one trains data from two different classes. In this application, there are 6 different classes relating to the 6 different poses described in Figure 1, plus an additional complementary class representing an indeterminate pose (with $k = 7$ we have 21 classifiers). In practice one-against-one method is one of the more suitable strategies for multi-classification problems [8]. After training data from the *i*th and the *j*th classes, the following voting strategy is used: if a binary classifier says that a given sample x is associated to class j,

then the vote for the class j is incremented by one. Finally, sample x is predicted to be in the class with the largest vote - this approach is referred as the *max-wins* strategy [21]-.

5 Results

We tested our system in a database of hand poses which was built up in the following way: We recorded 6 videos about 600 frames each (24 seconds long). Each video consisted of a sequence of hands performing the movements associated to only one specific pose. Next, the experts labelled each frame as pose 1 to 6, using the label 7 for the indeterminate pose, and those frames in which the experts did not agree were rejected. Following, the selected frames were analyzed by the ROI segmentation procedure described in Section 2, and all those frames in which the ROI could not be detected by the system because the hands were not joined -basically at the start and at the end of the video sequences- were rejected. The remaining frames -about a half of the original data set- constitute the training set with which the SVM ensemble was trained. The test set consisted of two videos of about 1 minute long in which the different poses were randomly visualized. The 3 experts performed a labelling of all the frames for these 2 videos, and all those frames in which the experts did not agree were rejected. In addition, all the frames in which a ROI was not segmented by the system were not taken into account. The remaining frames (1449/1924 for video 1 and 1025/1309 for video 2) constitute the test set.

Skin and non-skin reference histograms were obtained from the open source filtering Poesia project [17]. The project compiled 32^3 3D RGB histograms from the Compaq database and left them available in two files, one for skin pixels and other for non-skin pixels. Prior probabilities of skin and non-skin were assumed to be 0.4 and 0.6, which is a fair approach of the average region covered by the arms within the region of analysis. The segmented ROIs, from where features were extracted, were resized to a fixed size of 128×128 in order to have a normalised area of analysis. For the motion analysis we applied the Lucas and Kanade approach [13] with a threshold value $T_{ID} = 50$ and a window of 7 pixels for the average filter. We used the LIBSVM library [3] for building up the SVM classifier ensemble with radial basis function (RBF) kernels. All the single classifiers used a $\gamma = 3.2e^{-04}$ as the RBF parameter, which was obtained empirically.

We can see the results after applying multi-classification with HOG features and SVM classifier in Table 1. Although detection rate is low for the "other poses" case, which is reasonable taking into account that it is the class with higher variability, all the recommended poses are classified with detection rates greater than 85%. Three of them are classified with an accuracy greater than 91%, and the best classified class has a detection rate of 96.09%. The last row of Table 1 shows the inter-observer agreement as the percentage of coincident labels assigned by different experts to frames of the same class, which provides information about the intrinsic difficulty of each pose. The inter-observer agreement was calculated in the following way: We take all the M^1 frames labelled as class 1 by Expert 1 and Expert 2 and we count the number of coincidences $N_1^{1,2}$. The inter-observer agreement for Experts 1 and 2 in class 1 is calculated as the ratio $N_1^{1,2}/M^1$. We repeat the same procedure for Experts 2 and 3, and for Experts 1 and 3, and we average the results (this same scheme is applied for the rest of the classes).

The results shown in Table 1 appear to confirm that the proposed approach behaves in a similar way as the experts: Those classes which show the highest detection rate, as in the

Data sets	**Pose 1**	**Pose 2**	**Pose 3**	**Pose 4**	**Pose 5**	**Pose 6**	**Others**
Training Data Set	638	780	594	977	902	1073	587
Test Data Set	488	414	360	456	477	358	304
Detection rate	**86.07%**	**91.55%**	**94.72%**	**89.25%**	**86.37%**	**96.09%**	**61.84%**
Inter-observer agr.	94.59%	97.71%	97.20%	97.53%	98.46%	97.43%	82.98%

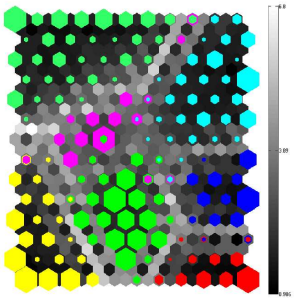
Table 1: Detection rate for the different classes

case of *Pose* 6, tend to show a high expert agreement, and viceversa -both low agreement and detection rate in *Pose* 7-, perhaps with an exception in the case of *Pose* 5. This fact can be analysed in the distribution table of Figure 6 (a), in which the element in the row i and column j represents the number of frames labelled by the human as *Pose* i which were classified by the system as *Pose* j. The distribution matrix shows how most of the misclassified samples of *Pose* 5 are voted for *Pose* 6 and the "other poses" class. However, the system appears to achieve a better generalization of *Pose* 6, since no misclassified frames are voted for *Pose* 5 in this case. Figure 6 (b) shows the cluster representation obtained using a self organizing map (SOM) [11] after a principal component analysis (PCA) holding 95% of the eigenvector spectrum, which produces a 2D mapping from the original feature space suitable for visualization purposes. Each sample in the data set is associated to one cell. The coloured cells in the SOM represents hit histograms of the different classes -the size of each cell being related to the number of samples in the cluster-, while the grey scale represents distances between cells. The SOM shows a clear cluster distribution, with several multi-class cells in the boundaries between clusters. This fact make us think that a simpler classification strategy specifically focusing on these regions could improve the classification results.

Figure 7 depicts an example of the single frame classification results in a given sequence. The detected class is shown with a number at the top left of each image. The system correctly detects when the hands are separate or joined. Even with a single frame approach the classifier correctly detects the transitions between poses, which are classified as the "other poses" class (a video sample of the system output can be obtained via web in the following link: http://www.cs.tcd.ie/VAMP/handwashing.avi).

Poses		SYSTEM						
		1	2	3	4	5	6	7
HUMAN	1	240	0	2	8	0	0	4
	2	0	198	0	0	14	0	0
	3	0	3	170	0	2	0	0
	4	1	0	0	292	2	4	0
	5	3	0	0	3	194	11	5
	6	3	0	0	0	0	143	2
	7	1	7	0	5	18	5	109

(a)



(b)

Figure 6: (a) Distribution table of the multi-class ensemble. (b) Cluster representation using SOMs. The grey scale represents distance between cells and each color is associated with a class-cluster. The size of each cell is related to the number of samples in the cluster.

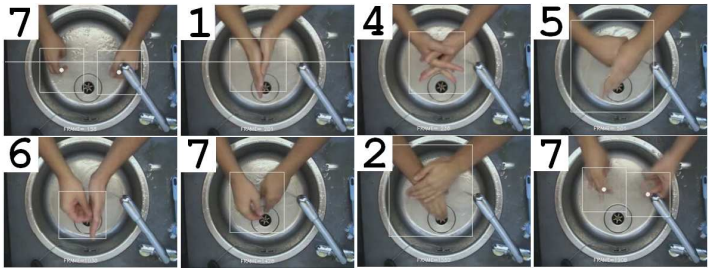


Figure 7: A sequence of frames with their associated pose class.

6 Conclusions and Future Work

In this work we presented a hand washing quality assessment system which can be used in hospitals, clinics, food processing industry, etc. One camera is placed over the sink focusing on the hand washing area. The system is implemented in a PC-based architecture and the complete algorithm runs at an average rate of 20 frames/sec. Hands/arms are segmented by means of skin and motion features. This segmentation process is robust against specular reflections due to steel sinks, illumination changes, etc. A SVM multi-class classification scheme using 21 binary classifiers, combined with histograms of oriented gradients locally distributed and normalised features, appear to yield very acceptable results. Hand washing poses that can be either done with left or right hand (symmetric poses) are correctly detected. The worst detected class is the one with higher variability (*other poses*) which has a detection rate of 61.84%. The minimum and maximum detection rate for the six poses are 86.07% (*Pose* 1) and 96.09% (*Pose* 6).

As a novel application, the outcomes shown in this contribution are encouraging results. However, several tasks are planned for the future: Hands/arms segmentation will be improved to detect the hands even if they are not moving. The search of an optimal tradeoff between the discrimination power of the feature vector and its size is an important issue in order to reduce the overall computation load in on-line systems. Single frame classification can be improved by adding a multi-frame validation process, in which stochastic models (hidden Markov models) could play an important role integrating sequence information. The use of additional features, particulary those based on motion by means of optical flow analysis and motion history gradients [2], are being investigated. Finally, although the multi-class SVM ensemble appears to show good classification results, the use of other alternatives with a lower computational cost are in our scope, looking forward to integrate this application into a feasible low cost FPGA platform.

References

[1] B.E Boser, I.M Guyon, and V.N Vapnik. A training algorithm for optimal margin classifiers. In *ACM Workshop on COLT*, pages 144–152, 1992.

[2] G.R. Bradsky and J.W. Davis. Motion segmentation and pose recognition with motion history gradients. *Machine Vision and Applications*, 13:174–178, 2002.

[3] C.C Chang and C.J Lin. LIBSVM: A library for support vector machines. Software available at http://www.csie.ntu.edu.tw/˜cjlin/libsvm, 2001.

[4] N Dalal and B Triggs. Histograms of oriented gradients for human detection. In *Proc. CVPR*, volume 2, pages 886–893, 2005.

[5] W. T. Freeman and M. Roth. Hand gesture recognition using orientation histograms. In *Proc. IEEE TENCON*, pages 1355–1368, 1999.

[6] B. Funt, K. Bernard, and L. Martin. A fast skin region detector. In *ESC Division Research*, 2005.

[7] J. Hoey, A. von Bertoldi, et al. Assisting persons with dementia during handwashing using a partially observable Markov decision process. In *Proc. ICVS*, 2007.

[8] C.W Hsu and C.J Lin. A comparison of methods for multi-class support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, 2002.

[9] M.J Jones and J.M. Rehg. Statistical color models with application to skin detection. *International Journal of Computer Vision*, 46(1):81–96, 2002.

[10] M. Klsch and M. Turk. Robust hand detection. In *Proc. IEEE CAFGR*, pages 614–619, 2004.

[11] T. Kohonen. *Self-Organized Maps*. Springer, Heidelberg Berlin, 1995.

[12] D. Lin. Spatio-temporal hand gesture recognition using neural networks. In *Proc. IEEE WCCI*, volume 3, pages 1794–1798, 1998.

[13] B.D. Lucas and Kanade T. An iterative image registration technique with an application to stereo vision. In *Proc. IUW*, pages 121–130, 1981.

[14] S. Marcel, O. Bernier, et al. Hand gesture recognition using input-output hidden markov models. In *Proc. IEEE CAFGR*, pages 456–461, 2000.

[15] MRSA. Methicillin-resistant staphylococcus aureus. Guidance for nursing staff. In *Royal College of Nursing*, 2005.

[16] V. Pavlovic, R. Sharma, and T Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Trans. PAMI*, 19(5):677–695, 1997.

[17] Poesia Filter. http://www.poesia-filter.org/.

[18] T. Starner and A. Pentland. Visual recognition of American sign language using hidden Markov models. In *Proc. WAFGR*, pages 189–194, 1995.

[19] M. Su, H. Huang, et al. Application of neural networks in spatio-temporal hand gesture recognition. In *Proc. IEEE WCCI*, pages 2116–2121, 1998.

[20] P Viola and M Jones. Robust real-time object detection. *Int. Journal on Computer Vision*, 2002.

[21] T. F Wu, C. J Lin, and R. C Weng. Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research*, 5:975–425, 2004.

Towards Real-Time Traffic Sign Recognition by Class-Specific Discriminative Features

Andrzej Ruta Yongmin Li
Xiaohui Liu
School of Information Systems, Computing and Mathematics
Brunel University, Uxbridge, Middlesex UB8 3PH, UK
{Andrzej.Ruta, Yongmin.Li, Xiaohui.Liu}@brunel.ac.uk

Abstract

Real-time road sign recognition has been of great interest for many years. This problem is often addressed in a two-stage procedure involving detection and classification. In this paper a novel approach to sign representation and classification is proposed. In many previous studies focus was put on deriving a set of discriminative features from a large amount of training data using global feature selection techniques e.g. Principal Component Analysis or AdaBoost. In our method we have chosen a simple yet robust image representation built on top of the Colour Distance Transform (CDT). Based on this representation, we introduce a feature selection algorithm which captures a variable-size set of local image regions ensuring maximum dissimilarity between each individual sign and all other signs. Experiments have shown that the discriminative local features extracted from the template sign images enable minimum-distance classification with error rate not exceeding 7%.

1 Introduction

Recognition of traffic signs has been a challenge problem for many years and is an important task for the intelligent vehicles. Although the first work in this area can be traced back to the late 1960's, only in the 1990's, when the problems of intelligent navigation and driver's safety attracted worldwide attention, significant advances were made. Nevertheless, there is still an apparent gap between what human and machine can do, making the attentive driver an irreplaceable guarantor of safety in the traffic environment.

Road signs have unique properties distinguishing them from the multitude of other outdoor objects. These properties were benefited from in numerous attempts to build an efficient detection and recognition system. In the majority of published work a two-stage sequential approach was adopted, aiming at locating the regions of interest first, and subsequently passing them to the classifier [1, 2, 3]. To detect possible sign candidates traditionally colour information is extracted [1, 2], followed by the geometrical edge [1, 4] or corner analysis [2]. Alternative approaches utilise distance transform [5] or neural networks [6]. In several studies the geometrical tracking aspect was given consideration [1, 6, 7]. However, reliable prediction of the geometrical properties of signs from a moving vehicle is complex in general as the vehicle's manoeuvres are enforced by the

actual traffic situation and therefore cannot be apriori known. To overcome this problem, the above approaches impose simplified motion model, e.g. assuming constant velocity. In the classification stage a pixel-based approach is often adopted and the class of the detected sign is determined by the cross-correlation template matching [1] or neural network [2]. Feature-based approach is used for instance in [3]. More recently, Bahlmann et al. [9] adopted the ideas of Viola and Jones [8] to detect traffic signs based on the colour-sensitive Haar wavelet features and AdaBoost framework. In the classification stage, assuming Gaussian class distribution and the independence of consecutive frame observations, Bayes classifier is used to fuse the individual observations over time. Only 6% error rate is reported using this method. Paclík et al. [10] introduced a different strategy built upon the claim that a candidate sign can be represented as a set of similarities to the stored prototype images. For each class similarity assessment is made with respect to a different set of local regions refined in the training process.

In this work we have developed a two-stage road sign detection and classification system. Figure 1 shows an example frame from video input with a road sign detected and recognised. More specifically, our detector is a form of well-constrained circle/regular polygon detector introduced in [4], augmented with the appropriate colour pre-filtering. In the classification stage, motivated by [10], we introduce a novel feature selection algorithm built on top of the Colour Distance Transform (CDT) image representation. We show that although our algorithm generates sign descriptors of variable dimensionality, individual classification scores can be made directly comparable due to the global selection criterion used. In consequence the proposed method seems to be a more natural way of discrimination among signs, as not the same amount of information is necessary to tell different classes apart. The rest of this paper is organised as follows: In section 2 traffic sign detection and tracking are briefly described. Sections 3 and 4 discuss the main contributions of this work, discriminative feature selection and sign classification. Section 5 presents experimental results on the real traffic video sequences. Finally, conclusions are drawn in section 6.

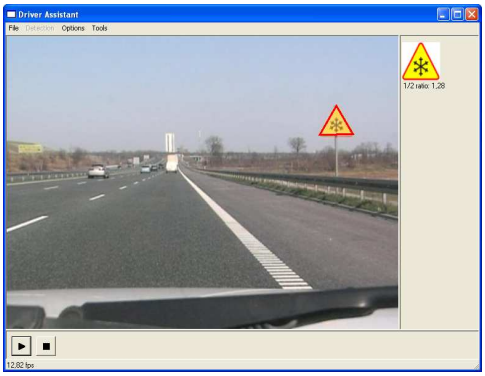


Figure 1: Screenshot from our traffic sign recognition system in action.

2 Sign Detection and Tracking

Our road sign detector is triggered every fixed number of frames to capture new candidates emerging in the scene. It makes use of the apriori knowledge about the model signs, uniquely identified by their general shape, colour and contained ideogram. Based on the first two properties four sign categories coinciding with the well-known semantic families are identified: instruction (blue circular), prohibitive (red circular), cautionary (yellow triangular), and informative (blue square) signs. As we believe the shape and boundary colour of a sign are sufficient visual cues to locate the candidates reliably, the proposed

detector operates on the colour gradient and edge maps of the original video frames. Furthermore, it uses a generalisation of Hough Transform introduced in [4], which is motivated by the fact that the targeted objects are all instances of equiangular polygons, including circles that can be though of as such polygons with the infinite number of sides.

Original regular polygon transform is augmented with the appropriate image pre-processing intended to enhance edges of specific colour. For each RGB pixel $\vec{x} = [x_R, x_G, x_B]$ and $s = x_R + x_G + x_B$ a simple colour enhancement is provided by a set of transformations:

$$\begin{array}{l} f_R(\vec{x}) = \max(0, \min((x_R - x_G)/s, (x_R - x_B)/s)) \\ f_B(\vec{x}) = \max(0, \min((x_B - x_R)/s, (x_B - x_G)/s)) \\ f_Y(\vec{x}) = \max(0, \min((x_R - x_B)/s, (x_G - x_B)/s)) \end{array} \quad . \tag{1}$$

Transforms defined in (1) effectively extract the red, blue, and yellow image fragments. In the resulting images colour-specific edge maps are extracted by a simple filter which for a given pixel picks the highest difference among the pairs of neighbouring pixels that could be used to form a straight line through the middle pixel being tested. Obtained values are further thresholded and only in the resulting edge pixels values of directional and magnitude gradient are calculated. This technique is adequate to our problem as it enables a quick extraction of edges and avoids expensive computation of the whole gradient magnitude map which, with the exception of the sparse edge pixels, is of no use to the shape detector. For a given pair of gradient and edge images associated with colour c, appropriate instances of the Loy & Barnes's detector are run to yield a set of possible sign shapes. For instance, for a "blue pair" a circular shape detector is triggered to search for the blue instruction signs, e.g. "turn left" or "turn right", and a square detector is run to detect potential square information signs, e.g. "pedestrian crossing" or "parking place". As each found candidate has known shape and border colour c, detector serves as a pre-classifier reducing the number of possible templates to analyse in the later stage to the ones contained in either category. When signs are in the cluttered background, a number of false candidates may be produced. To address this issue, an additional step is taken to verify the presence of colours appropriate for the just found category.

Once a candidate sign is detected, it is unnecessary to seek it in the consecutive frames in every possible location. Assuming motion with constant velocity along the optical axis of the camera, we have employed a Kalman filter [11] to track a sign detected in a previous frame of an input video. The state of the tracker is defined by (x, y, s), where x, y are coordinates of the sign's centre in the image, and s is the scale factor to the standard sign templates. In the current implementation we use the mean and variance estimates from the Kalman filter to locate the centre and the size of the local search region in the next video frame. Therefore, computation has been significantly reduced compared to the exhaustive search over the whole image.

3 Image Representation and Feature Selection

Selecting an optimal feature set for a large number of template images is a non-trivial task. We have experimented with several techniques such as Principal Component Analysis and AdaBoost. Aiming at retrieving the global variance of a whole data set, PCA is not capable of capturing features critical to the individual templates. AdaBoost on the other hand, although generating efficient classifiers, is not entirely convincing in terms of the fixed cardinality of the feature set being extracted. Clearly, certain signs are very dis-

tinctive and analysis of only a few small regions suffice to distinguish them even among tens of others. Meanwhile, there are groups of very similar signs that look tightly clustered, even in a highly multidimensional feature space. This complex nature of similarity between templates raises a question whether there is sufficient justification for classifying the signs based on the same set of features.

Motivated by [10], we propose here an algorithm that relaxes the above limitation by extracting for each model sign a limited number of local image regions in which it looks possibly the most different from all other templates residing in the same category. The same discriminative regions are further used to compare a video frame image with the templates and make a reliable on-line classification. Below we first outline the process of converting the raw bitmap images into a more suitable discrete-colour representation. Second, we introduce the notion of local image region and dissimilarity. Finally, the implementation of the discriminative region selection algorithm is discussed.

3.1 Colour Discretisation

Detected sign images come as rectangular regions containing the target object and, depending on its shape, also background fragments, as depicted in Fig. 2. In order to prepare the candidate for classification, the image is first scaled to a common size, typically 60×60 pixels. Undesirable background regions are then masked out using the information about the object's shape and orientation provided by the detector [4]. It is important to note that the full colour spectrum is far more than necessary to identify the object, as the signs contain only up to four distinctive colours per category. Therefore, the candidate images are finally subject to on-line colour discretisation according to the category-specific colour models learned off-line from a set of training images as follows.

For each category of signs a number of frames are picked randomly from the real video sequences depicting the respective signs. Then, the Expectation Maximisation algorithm [12] is employed to estimate an optimal Gaussian Mixture model for each colour specific to this category. The procedure is restarted several times for the increasing number of randomly initialised Gaussian components to refine the estimation. The best model in terms of the mean data likelihood is selected. To speed up the on-line segmentation, off-line learned models are used by a Bayes classifier to assign the appropriate colour to each possible RGB triple, yielding a look-up table with 255^3 entries for each category. Sample results of the on-line colour discretisation are illustrated in Fig. 2.

Figure 2: Sample images obtained by sign detector before (above) and after (below) background masking and colour discretisation; 2 bits suffice to encode colours in each image.

Along with the observed candidate sign images also the model images are discretised. However, as they already contain ideal colours, discretisation merely aims here at collapsing the physical 24-bit bitmap representation to the 2-bit image with the specific colour indices encoded. A set of thresholds is applied to the templates in Hue-Saturation-Value space to complete this task. Furthermore, for each discrete colour present in the resulting

images a separate distance transform [13] is computed, giving output similar to this shown in Fig. 3. In DT computation pixels of a given colour are simply treated as feature pixels and all the remaining ones as non-feature pixels. $(3,4)$ Chamfer metric [14] is used to approximate Euclidean distance between the feature pixels. To emphasise a strong relation to colour, we call this variant of DT a Colour Distance Transform (CDT).

(a) (b) (c) (d)

Figure 3: Colour Distance Transform images: original discrete colour image (a), black CDT (b), white CDT (c), red CDT (d); darker regions denote shorter distance.

3.2 Discriminative Local Regions

The space of local regions is obtained by covering the template image with a regular grid. Within each region r_k dissimilarity between the images I and J can be calculated using the discrete-colour image of I and CDT images of J by averaging the pixel-wise distances:

$$d_{r_k}(I,J) = \frac{1}{m}\sum_{t=1}^{m} d_{CDT}(I(p_t),J(p_t)) \ , \tag{2}$$

where for each of m pixels p_t contained in the region, distance $d_{CDT}(I(p_t),J(p_t))$ is picked from the appropriate CDT image of J, depending on the colour of this pixel in I. Let us also denote by $d_{\mathbf{S}}(I,J)$ and $d_{\mathbf{S},\mathbf{W}}(I,J)$ a normal and weighted average local dissimilarities between the images I and J computed over regions $r_k \in \mathbf{S}$ (weighted by $w_k \in \mathbf{W}$):

$$d_{\mathbf{S}}(I,J) = \frac{1}{M}\sum_{k=1}^{M} d_{r_k}(I,J) \ , \tag{3}$$

$$d_{\mathbf{S},\mathbf{W}}(I,J) = \frac{\sum_{k=1}^{M} w_k d_{r_k}(I,J)}{\sum_{k=1}^{M} w_k} \ . \tag{4}$$

Obviously, as CDT images for the model signs are pre-computed, any on-line local-region comparisons between the observed and template images can run extremely fast.

3.3 Region Selection Algorithm

Assuming pre-determined category of signs $C = \{T_i : i = 1,\ldots,N\}$ and a candidate image x_j, our goal is to determine the class of x_j by maximising posterior:

$$p(T_i|x_j,\theta_{\mathbf{i}}) = \frac{p(x_j|T_i,\theta_{\mathbf{i}})p(T_i)}{\sum_{i=1}^{N} p(x_j|T_i,\theta_{\mathbf{i}})} \ . \tag{5}$$

Our objection to using a uniform feature space for classification makes us envisage different model parameters $\theta_{\mathbf{i}} = (\mathbf{I_i},\mathbf{W_i})$ for each template T_i. $\mathbf{I_i}$ denotes an indexing variable determining the set $\mathbf{S_i}$ of regions to be used and $\mathbf{W_i}$ is a vector of relevance corresponding to the regions $r_k \in \mathbf{S_i}$ selected by $\mathbf{I_i}$. In order to learn the best model parameters $\theta_{\mathbf{i}}^*$, the following objective function is maximised:

$$O(\theta_{\mathbf{i}}) = \sum_{j\neq i} d_{\mathbf{S_i}}(T_j,T_i) \ . \tag{6}$$

In other words, the regions best characterising a given sign are obtained through maximisation of the sum of local dissimilarities between this sign's template and all the remaining signs' templates. In presence of model images only, each term $d_{\mathbf{S_i}}(T_j, T_i)$ as a function of the number of discriminative regions is necessarily monotonically decreasing. As a result, typically there would be just a few good regions maximising (6). In practice, such sign descriptors are unlikely to work well for the noisy video frames where more support in terms of the number of image patches to match is required to make a reliable discrimination. Our objective function, as described in the algorithm 1, is hence iteratively degraded up to the specified breakpoint, yielding a representation which is richer and thus more trustworthy in a real-data scenario.

Algorithm 1 Discriminative local region selection algorithm

input: sign category $C = \{T_j : j = 1,\ldots,N\}$, target template index i, region pool $R = \{r_k : k = 1,\ldots,M\}$, dissimilarity threshold t_d
output: target set F_i of regions with associated weights

initialise an array of region weights $W = \{w_k : w_k = 0, k = 1,\ldots,M\}$
for each template $T_j \in C$, $j \neq i$ **do**
  find region $r_{j,1}$ such that $d_{r_{j,1}}(T_j, T_i) = \max_k d_{r_k}(T_j, T_i)$
  initialise ordered region list $F_j = [(r_{j,1}, w_{j,1})]$, where $w_{j,1} = 1$
  initialise remaining feature pool $P_j = R \setminus \{r_{j,1}\}$ and region counter $l = 1$
  while not STOP **do**
    increment region counter $l = l + 1$
    for each region $r_k \in P_j$ **do**
      construct a region list $S_k = F_j + r_k$
      pick region $r_{j,l}$ maximising $d_{S_k}(T_j, T_i)$
      set weight of the found region to $w_{j,l} = d_{r_{j,l}}(T_j, T_i)/d_{r_{j,1}}(T_j, T_i)$
    end for
    add pair $(r_{j,l}, w_{j,l})$ to the selected region list $F_j = F_j + (r_{j,l}, w_{j,l})$
    update the remaining region pool $P_j = P_j \setminus \{r_{j,l}\}$
    if $d_{F_j}(T_j, T_i) < t_d d_{r_{j,1}}(T_j, T_i)$ **then**
      STOP = true
    end if
  end while
  for each region r_k such that $(r_k, w_{k,j}) \in F_j$ **do**
    update found region weights $w_k = w_k + w_{k,j}$
  end for
end for
build target region set $F_i = \{(r_k, w_k) : w_k > 0\}$

Similarly to Paclík et al. [10], in the model training stage we have adopted elements of a sequential forward search strategy, a greedy technique from the family of floating search methods [15]. However, both approaches differ significantly in the two main aspects. First, we think that learning the signs from the real-life images is counter-intuitive as the publicly available templates characterise respective classes fully. Second, we believe that the possible within-class appearance variability may well be accounted for by a robust distance metric, as the one introduced in (2-4), instead of being learned. Our implementation then picks a given template sign and compares it to each of the remaining templates. In each of such comparisons the algorithm loops until the appropriate number

of local regions are found. It should be noted that at a given step of the loop the most dissimilar region is fixed and removed from the pool of available regions. Moreover, at the k-th step the distance between the considered image and the image being compared to is measured with respect to the joint set comprised of the new k-th region and all previously found regions. At the end of the loop an ordered list of regions is produced, sorted by their decreasing discriminative power. Each pairwise region set build-up is controlled by a global threshold, t_d, specifying the minimum allowed dissimilarity between any pair of templates being compared as a percentage of the maximum possible dissimilarity, i.e. the one for just a single most discriminative region. Such a definition of STOP criterion ensures that the same amount of dissimilarity between any pair of templates is incorporated in the model. This in turn allows us to treat different sign classes as directly comparable, irrespective of their actual representation. The final set for each class is constructed by merging the pair-specific subsets which is reflected in the region weights carrying the information on how often and with what contribution each particular region was selected.

For each sign the above algorithm yields a set of its most unique regions. It should be noted that in the final step, depending on the actual dissimilarity threshold specified, certain number of regions will be found completely unused, and hence discarded. An example of our feature selector's output is depicted in Fig. 4. Obtained discriminative region maps clearly show that different signs are best distinguishable in different fragments of the contained pictogram. It can also be seen that although the same value of global parameter t_d was used, different numbers of meaningful regions remained.

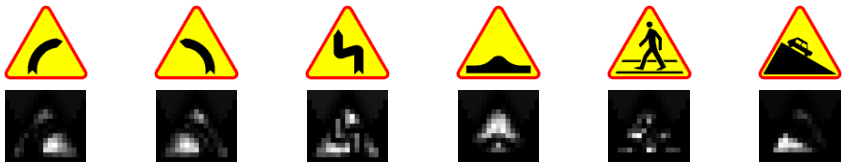


Figure 4: Sample triangular template images (above), and discriminative regions obtained for parameter $t_d = 0.7$ (below); brighter regions correspond to the higher dissimilarity.

4 Temporal Classifier Design

A road sign classifier distinguishes between the sign classes contained in a category pre-determined in the detection stage, based on the discriminative feature representation unique for each particular sign. For simplicity two assumptions are made: 1) the dissimilarity between each sign and all other same-category signs is Gaussian-distributed in each local region and independent of the dissimilarities in all other regions characterising this sign, and 2) class priors $P(T_i)$ are equal. In such a case Maximum Likelihood theory allows us to relate the maximisation of likelihood $p(x_j|T_i,\theta_\mathbf{i})$ to the minimisation of weighted distance $d_{\mathbf{S_i},\mathbf{W_i}}(x_j,T_i)$. Therefore, for a known category $C = \{T_i : i = 1,\ldots,N\}$, and observed candidate x_t at time t, the winning class $L(x_t)$ is determined from (5):

$$L(x_t) = \arg\max_i p(x_j|T_i,\theta_\mathbf{i}) = \arg\min_i d_{\mathbf{S_i},\mathbf{W_i}}(x_t,T_i) \quad , \tag{7}$$

where $\mathbf{S_i}$ and $\mathbf{W_i}$ contain the regions and their weights learned in the training stage.

When a series of observations from a video sequence is available, it is reasonable to integrate the classification results through the whole sequence over time, instead of per-

forming individual classifications. Hence, at a given time point t our temporal integration scheme attempts to incorporate all the observations made since the sign was for the first time detected until t. Denoting observation relevance by $q(t)$ and assuming independence of the observations from consecutive frames, the classifier's decision is determined by:

$$L(X_t) = \arg\min_i \sum_{k=1}^{t} q(t) d_{\mathbf{S_i},\mathbf{W_i}}(x_k, T_i) \ . \tag{8}$$

We have observed that the signs detected in the early frames are inaccurate and contain blended pictograms due to the low resolution. Also as colours tend to be paler when seen from the distance, previously discussed colour discretisation exposes severe limitations, unless performed for later frames depicting candidate sign already grown in size. To address this problem, we adopt the exponential observation weighting scheme from [9] in which relevance $q(t)$ of observation x_t depends on the candidate's age (and thus size):

$$q(t) = b^{t_0 - t} \ , \tag{9}$$

where $b \in (0,1]$ and t_0 is the time point when the sign is for the last time seen.

5 Experiments

To evaluate our traffic sign recognition system, experiments were performed on the real data collected on Polish roads. Sample video sequences were acquired from a moving car with a DV camcorder mounted in front of the windscreen, and subsequently divided into short clips for off-line testing. Video content depicts the total of 144 signs and includes urban, countryside, and motorway scenes in natural daytime lightning, with numerous signs appearing in shade and in cluttered background. Table 1 illustrates obtained results.

	t_d	RC (55)	BC (25)	YT (42)	BS (13)	overall (135)
detected	–	85.2%	100.0%	98.3%	89.7%	93.8%
recognised	0.97	95.7%	93.9%	86.4%	85.7%	92.0%
	0.9	95.7%	97.0%	91.5%	91.4%	93.3%
	0.5	95.7%	90.9%	84.7%	82.9%	87.3%
	best	95.7%	97.0%	91.5%	91.4%	93.3%

Table 1: Recognition performance for different values of dissimilarity threshold t_d and temporal weight base $b = 0.8$; the number of classes in each category: red circles (RC), blue circles (BC), yellow triangles (YT), and blue squares (BS) is given in parentheses and the best classification rate is highlighted.

As seen in Tab. 1, obtained real-time classification error rate does not exceed 7%, making our method comparable to the recently published ones [9, 10]. However, it should be noted that our template database contains significantly more signs than in any of the previous studies. Direct comparison with the respective algorithms is not possible as neither the test data nor the details of its acquisition are made available. Repetitions of the experiment for different values of dissimilarity threshold revealed that for each category of signs the optimal classifier's performance is achieved for a close to maximum value of this threshold. The following observation is vital at this point. The optimal threshold for each category must strike a balance between the two: maximising template signs'

separability and the reliability of the obtained dissimilarity information in the real-data context. Very high threshold values lead to the separation of a very few good regions for a particular model sign, however such sparse information may not be sufficiently stable to classify correctly a possibly distorted, blurred, or occluded object in a video frame. Very low threshold values on the other hand introduce information redundancy by allowing image regions that contribute little to the uniqueness of a given sign. In a resulting feature space signs simply look more similar to one another and are hence more difficult to tell apart at an additional cost of more intense computation.

In terms of detection, most of failures were caused by the insufficient contrast between a sign's boundary and the background, especially for pale-coloured and shady signs. In a few cases this low contrast was caused by the poor quality of the physical target objects rather than their temporarily confusing appearance. Single detection errors emerged when two signs were mounted closely on one pole. In this particular situation candidate objects may be confused with each other, as the local search region of one candidate always contains at least part of its neighbour. Detection proved to be the computationally most expensive part of the system, however processing speed of the entire algorithm including classification is 10-20 fps on a standard PC, depending on the actual difficulty of the scene.

After closer investigation we have observed that approximately one in three classification errors resulted from confusion between nearly identical classes, e.g. pedestrian crossing and bicycle crossing. Differences between such signs were found difficult to capture, resulting sometimes in the correct template receiving the second best score. Colour segmentation appeared to be resilient to variations of illumination, leading directly to failure in only a few cases when the signs were located in a very shady area or were themselves of poor quality. This can be a proof of usefulness of Gaussian Mixture colour modelling. Remaining failures can be attributed to the limitations of the detector. Although the smooth distance metric neutralises the inaccurate detection effects to a large extent, it is of little help in certain situations, typically in two:

1. Some signs' ideograms consist of edges that may actually be easier to detect than the boundary. This may cause detected shape to appear clipped.
2. Signs very close to the camera being distorted in perspective projection usually receive sufficient score in the detector's accumulator space. These signs are yet still detected as regular shapes, resulting in the inaccurate shape estimation.

As indicated in the previous section, delimitation of sign's contour and subsequent colour discretisation in the early video frames are usually less accurate. Extensive experiments have shown that frequently the correct decision is developed by the classifier from just a few last frames where the sign's shape and colours are the most unambiguously detected. This fact provides a good justification for our exponential observation weighting used to promote the most recent measurements. Apparently, the classification accuracy with weighting enabled is by 10-20 % higher, depending on the weight base b used.

6 Conclusions

In this paper we have introduced a novel image representation and discriminative feature selection method for road sign recognition where a large number of classes are involved. It was shown that on top of a Colour Distance Transform (CDT) representation highly discriminative sign descriptors can be extracted based on the principle of dissimilarity maximisation. With these descriptors available, a conventional classifier can compete

with the state-of-the-art sign recognition systems, processing the input video sequences in close to real time. In comparison to the previous studies, our method seems attractive in three aspects. First, feature selection is performed directly on the publicly available template sign images. Second, each template is treated on an individual basis which is reflected in the number, position, and importance of the local image regions extracted in order to achieve a desired level of dissimilarity from the remaining templates. Finally, by using CDT we have shown that the proposed description of signs can be extended from model images to the real video frames as the resulting distance measure is made smoother and thus more resistant to various types of noise typically affecting the video content.

References

[1] Piccioli, G., De Micheli, E., Parodi, P., and Campani, M. A robust method for road sign detection and recognition. *Image and Vision Computing*, 14(3):209-223 (1996)

[2] de la Escalera, A., Moreno, L. E., Salichs, M. A., and Armingol, J. M. Road traffic sign detection and classification. *IEEE Trans. on Industrial Electronics*, 44(6):848-859 (1997)

[3] Paclík, P., Novovicova, J., Pudil, P., and Somol, P. Road Sign Classification using the Laplace Kernel Classifer. *Pattern Recognition Letters*, 21(13-14):1165-1173 (2000)

[4] Loy, G., Barnes, N., Shaw, D., and Robles-Kelly, A. Regular Polygon Detection. *In Proc. of the 10th IEEE Int. Conf. on Computer Vision*, 1:778-785 (2005)

[5] Gavrila, D. Multi-feature Hierarchical Template Matching Using Distance Transforms. *In Proc. of the IEEE Int. Conf. on Pattern Recognition*, Brisbane, Australia, 439-444 (1998)

[6] Fang, C-Y., Chen, S-W., and Fuh, C-S. Road-Sign Detection and Tracking. *IEEE Trans. on Vehicular Technology*, 52(5):1329-1341 (2003)

[7] Miura, J., Kanda, T., and Shirai, Y. An active vision system for real-time traffic sign recognition. *In Proc. of the IEEE Conf. on Intelligent Transportation Systems*, Darborn, MI, USA, 52-57 (2000)

[8] Viola, P., and Jones, M. Robust Real-time Object Detection, *International Journal of Computer Vision*, 57(2):137-154 (2004)

[9] Bahlmann, C., Zhu, Y., Ramesh, V., Pellkofer, M., and Koehler, T. A System for Traffic Sign Detection, Tracking and Recognition Using Color, Shape, and Motion Information. *In Proc. of the IEEE Intelligent Vehicles Symposium*, 255-260 (2005)

[10] Paclík, P., Novovicová, J., and Duin, R. P. W. Building Road-Sign Classifiers Using a Trainable Similarity Measure. *IEEE Trans. on Intelligent Transportation Systems*, 7(3):309-321 (2006)

[11] Kalman, R. E. A New Approach to Linear Filtering and Prediction Problems. *Trans. of the ASME - Journal of Basic Engineering*, 82:35-45 (1960)

[12] Dempster, A., Laird, N., and Rubin, D. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*(B), 39(1):138, (1977)

[13] Borgefors, G. Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*, 34(3):344-371 (1986)

[14] Akmal Butt, M. and Maragos, P. Optimum design of chamfer distance transforms. *IEEE Trans. on Image Processing*, 7(10):1477-1484 (1998)

[15] Pudil, P., Novoviová, J., and Kittler, J. Floating search methods in feature selection. *Pattern Recognition Letters*, 15(11):1119-1125 (1994)

Unsupervised extraction of coherent regions for image based rendering

Jesse Berent and Pier Luigi Dragotti
Electrical and Electronic Engineering Department
Imperial College London
Exhibition Road, SW7 2AZ, UK
{jesse.berent, p.dragotti}@imperial.ac.uk

Abstract

Image based rendering using undersampled light fields suffers from aliasing effects. These effects can be drastically reduced by using some geometric information. In pop-up light field rendering [18], the scene is segmented into coherent layers, usually corresponding to approximately planar regions, that can be rendered free of aliasing. As opposed to the supervised method in the pop-up light field, we propose an unsupervised extraction of coherent regions. The problem is posed in a multidimensional variational framework using the level set method [16]. Since the segmentation is done jointly over all the images, coherence can be imposed throughout the data. However, instead of using active hypersurfaces, we derive a semi-parametric methodology that takes into account the constraints imposed by the camera setup and the occlusion ordering. The resulting framework is a global multidimensional region competition that is consistent in all the images and efficiently handles occlusions. We show the validity of the method with some captured multi-view datasets. Other special effects by coherent region manipulation are also demonstrated.

1 Introduction

The notion of plenoptic function [1] is central to image based rendering (IBR). It was introduced by Adelson and Bergen as a way of parameterizing the visual information available from any viewpoint, in any direction at any time and wavelength. IBR is essentially the problem of sampling and interpolating the plenoptic function. It is far from trivial to deal with the seven dimensions therefore several parameterizations have been proposed that reduce the dimensionality. The light field [13] or the lumigraph [9] are popular four dimensional parameterizations that limit the viewing space to a bounding box. Each light ray is usually parameterized by its intersection with two planes namely the image plane (u,v) and the camera location plane (s,t). Further reducing the dimensionality to three leads to a simplified light field [7] where the cameras are constrained to line. This setup is also known as the epipolar plane image (EPI) volume [4].

In conventional light field rendering, the data is highly sampled and a single depth plane is enough to render views with little or no aliasing. However, more geometric information is needed when the data is more sparsely sampled and occlusions occur. The tradeoff between geometry and sampling density was studied in [6]. Some IBR methods use complex geometry for rendering single objects such as in [9, 5]. However, these

models are sometimes difficult to obtain in scenes containing numerous objects and occlusions.

In pop-up light field rendering [18], the multi-view data is segmented into coherent layers. Each coherent region is represented with a collection of corresponding layers in each of the input images. These coherent layers are defined as regions in the light field that can be individually rendered free of aliasing. In [18], the segmentation is performed in a supervised fashion where the contours of layers are semi-manually defined on a key frame. These contours are then propagated to all the views using a user defined planar depth map. By performing the extraction in this way, the layers across all the image are coherent which enables a consistent rendering.

In this paper, we propose an unsupervised multidimensional analysis to extract coherent regions in multi-view data. Our method is based on the fact that objects in the scene carve out EPI-tubes [8] in EPI volumes and hypervolumes in light fields. Our approach therefore imposes coherence in all dimensions in order to extract these volumes or hypervolumes and generates a regular and approximate decomposition of the scene. The method presented is a novel multidimensional variational framework that explicitly takes into account the structure of the data. That is, the contours of the hypervolumes in the (u,v) dimensions (i.e. the images) are extracted using the level set method [16] which is a non parametric approach. However the dependencies between views (i.e. the two other dimensions (s,t)) and occlusions are constrained by the camera setup. They can therefore easily be parameterized. The resulting framework is a semi-parametric region competition that is global and hence uses all the available data jointly. This in turn insures that the segmentation is coherent across all the images and occlusions are efficiently and naturally handled. The main contributions of this work are the extraction of multiple coherent regions using a global variational framework and the formalization the method described in [3] using a more rigorous derivation of the gradient of the energy functional. Furthermore, the method is generalized to four dimensional light fields. Several other unsupervised layer extraction schemes have been proposed in the literature (e.g. [21, 2, 14, 22]). Not all these methods are scalable to higher dimensions and treat all the images equally. Moreover, most of these methods focus on the extraction of layers rather than the coherent volumes or hypervolumes carved out in multi-view data.

The paper is organized as follows: In Section 2 we study the shape constraints of the hypervolumes carved out by coherent regions in light fields. Section 3 derives a multidimensional variational framework based on active contours [12] for the extraction of the coherent regions and Section 4 shows some applications in image based rendering. In particular, view interpolation and some special effects are demonstrated. We conclude in Section 5.

2 Problem statement

Let $\vec{X} = (X,Y,Z)$ be a point in space, (u,v) be the image coordinates and (s,t) the position of the camera on a plane. Then, assuming a pinhole camera model [10], it can be shown that $\vec{X}$ is mapped onto the light field $L(u,v,s,t)$ according to

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \mapsto \begin{pmatrix} u \\ v \\ s \\ t \end{pmatrix} = \begin{pmatrix} X/Z - s/Z \\ Y/Z - t/Z \\ s \\ t \end{pmatrix}, \tag{1}$$

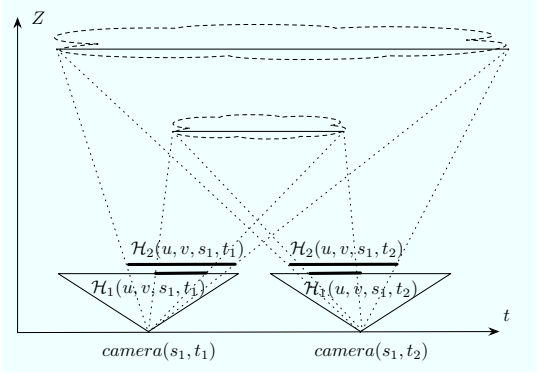


Figure 1: Foreground and background regions in the scene are projected on to the image planes (u,v) as a function of the camera location (s,t). Both regions carve out hypervolumes $\mathscr{H}_1$ and $\mathscr{H}_2$ in the light field $L(u,v,s,t)$. The intersection of these hypervolumes with a plane in $(s,t) = constant$ corresponds to the view of the layer in that image.

where we assume that the focal length is unity. That is, a point in space is mapped onto a four dimensional trajectory in the light field hypervolume. Note that this trajectory as a function of (s,t) is a line. Under the assumption of Lambertian surfaces, the intensity along the line will remain fairly constant unless there is an occlusion. The effect of specularities in the EPI setup has been studied in [8, 20] although we will not take these effects into account here. Objects or layers that are made of neighboring points in space will be mapped onto neighboring trajectories. Gathering these trajectories generates a multidimensional hypervolume $\mathscr{H}_n$ in which the information is highly regular. Assume approximately constant depth layers in the scene. Then the boundary of the hypervolume $\vec{\Gamma}_n = \partial\mathscr{H}_n$ carved out by the layer in the light field can be parameterized as

$$\vec{\Gamma}_n(l,s,t) = \begin{pmatrix} u_n(l) - s/Z_n \\ v_n(l) - t/Z_n \\ s \\ t \end{pmatrix}, \tag{2}$$

where $\vec{\gamma}_n(l) = (u_n(l), v_n(l))$ is the contour of the layer on the image in $s = t = 0$ and Z_n is the depth. As illustrated in Figure 1, the intersection of the hypervolume with an image in $(s,t) = constant$ is the layer on that image. Note that, due to occlusions, the full hypervolume $\mathscr{H}_n$ is not always available. Therefore, we denote with $\mathscr{H}_n^{\perp}$ the available coherent region. From (1), we see that the slopes of the lines constituting the region are inversely proportional to the depth of the points. Hence a line with a higher slope will always occlude a line with a smaller slope. Assuming the hypervolumes are ordered from front $(n = 1)$ to back $(n = N)$, we can write

$$\mathscr{H}_n^{\perp} = \mathscr{H}_n \cap \sum_{i=1}^{n-1} \overline{\mathscr{H}_i^{\perp}}, \tag{3}$$

where $\overline{\mathscr{H}_n^{\perp}}$ denotes the outside of the region and $\mathscr{H}_n^{\perp}$ is the hypervolume geometrically orthogonalized with the other hypervolumes occluding it (i.e. where i goes from 1 to $n-1$). The problem of extracting coherent regions in the light field consists in estimating

all the $\mathscr{H}_n^{\perp}$ using the available images. Note that the case of the simplified light field is straightforwardly obtained by posing $s = 0$. In the next section, we derive a fully unsupervised method to extract these hypervolumes in a global manner that takes advantage of the particular nature of the data.

3 A variational approach for extracting coherent regions in light fields

Since the seminal work of Kass et al. [12], active contours have been extensively used for numerous computer vision applications. Part of their success is due to the introduction of the level set method [16] which solved some of the issues such as numerical stability and topology dependance. Moreover, contrary to some other methods, the level set framework scales naturally to any number of dimensions [19]. This is a useful trait to analyze the data in light fields since it is parameterized with four dimensions.

Following this variational framework, we set the extraction of the coherent regions as an energy minimization problem. The initialization is performed in an unsupervised fashion by finding local directions in the EPIs and merging regions with similar slopes. The directions are found by locally estimating the angles of the EPI lines using a simple variance based method. The advantage of this approach consists in the fact that no reference image is used. The coherent regions $\mathscr{H}_n^{\perp}$ are then made dependent on an evolution parameter τ such that the derivative can be computed in order to evolve the hypervolumes in a steepest descent fashion. The total energy to minimize can be written in the form of a global region competition

$$E_{tot}(\tau) = \sum_{n=1}^{N} \int_{\mathscr{H}_n^{\perp}(\tau)} d_n(\vec{x}) d\vec{x},$$

where the scene is segmented into N regions, $\vec{x} = (u, v, s, t)$ and the $d_n(\vec{x})$ are descriptors measuring the consistency with a depth Z_n. One way to minimize the E_{tot} consists in iteratively evolving one hypervolume while keeping the others fixed. In this case, the energy for each iteration is written as

$$E_{tot}(\tau) = \int_{\mathscr{H}_n^{\perp}(\tau)} d_n(\vec{x}) d\vec{x} + \int_{\overline{\mathscr{H}_n^{\perp}}(\tau)} d_{out}(\vec{x}) d\vec{x},$$

where all the other regions are gathered in $\overline{\mathscr{H}_n^{\perp}}(\tau)$ and $d_{out}(\vec{x}) = d_i(\vec{x})$ when $\vec{x} \in \mathscr{H}_i^{\perp}$ for all $i \neq n$. Region based active contour methods such as [11, 19] have shown that the gradient of the energy in this case can be written as

$$\frac{dE_{tot}(\tau)}{d\tau} = \int_{\partial \mathscr{H}_n^{\perp}} [d_n(\vec{\sigma}) - d_{out}(\vec{\sigma})](\vec{v}_{\Gamma_n} \cdot \vec{n}_{\Gamma_n}) d\vec{\sigma},$$

where $\vec{v}_{\Gamma_n} = \frac{\partial \vec{\Gamma}_n}{\partial \tau}$ is the velocity, $\vec{n}_{\Gamma_n}$ is the outward unit normal vector, $d\vec{\sigma}$ is a differential surface element and $\cdot$ denotes the scalar product. Due to the occlusion ordering in (3), a foreground region evolving changes the background regions. That is, the evolution of $\mathscr{H}_n^{\perp}$ changes all the $\mathscr{H}_i^{\perp}$ where i goes from $n+1$ to N. Hence these occluded regions will contribute to the $\frac{dE_{tot}(\tau)}{d\tau}$ which leads to a competition. However, the other hypervolumes (i.e. $\mathscr{H}_i^{\perp}$ where i goes from 1 to $n-1$) are not affected by the shape changes in $\mathscr{H}_n^{\perp}$

and thus do not contribute to the derivative with respect to τ. Hence they will not compete. The steepest descent of the energy therefore yields the following partial differential equation

$$\vec{v}_{\Gamma_n} = [d_{out}(\vec{\sigma}) - d_n(\vec{\sigma})]\vec{n}_{\Gamma_n}$$

which will drive the evolution of the hypersurface in an unconstrained fashion. However, from Section 2, we know that the shape of the hypervolume in the light field is constrained. We can prove by doing an explicit derivation of the normal and velocity vectors using the parameterization in (2), that the normal velocity of the hypersurface $(\vec{v}_{\Gamma_n} \cdot \vec{n}_{\Gamma_n})$ projected onto the 2D subspace in $s = t = 0$ is equal to the normal velocity of the curve $(\vec{v}_{\gamma_n} \cdot \vec{n}_{\gamma_n})$ in one image[1]. The intuition behind this property is that given the fact that the region in space is approximately fronto-parallel, its contour will simply be a translated version of itself on all the images. The non fronto-parallel case is obtained by used a weighting term that compensates for the shape changes in the views. Using this constraint, the gradient of the energy can be rewritten as

$$\frac{dE_{tot}(\tau)}{d\tau} = \int_{\gamma_n} [D_n(l) - D_{out}(l)](\vec{v}_{\gamma_n} \cdot \vec{n}_{\gamma_n}) dl,$$

where the $D_n(l)$ and $D_{out}(l)$ are the original descriptors integrated over the line delimiting the boarder of the hypervolume. This leads to a new evolution equation for the 2D contour

$$\vec{v}_{\gamma_n} = [D_{out}(l) - D_n(l) - \mu\kappa]\vec{n}_{\gamma_n}, \tag{4}$$

where a smoothness term proportional to the curvature κ is added in order insure regular curves. The μ is a constant weighting factor determining the influence of the smoothness term. The advantage of such a manipulation consists in the fact that the constraints are fully taken into account (i.e. points are mapped onto lines) and it can be implemented as an active contour instead of an active hypersurface which reduces the computational complexity. We perform the overall energy minimization after initialization in three iterative steps: First, estimate the slopes of the lines in each individual hypervolume $\mathscr{H}_n^{\perp}$ using classical non-linear optimization methods. Second, update the occluded regions using the occlusion constraint in (3). Third, evolve each boundary $\vec{\gamma}_n$ individually using the evolution equation in (4) and the level set method. The algorithm is stopped when there is no significant decrease in the total energy.

Some results illustrating the coherent region extraction in simplified light fields are presented. These are easier to capture and more practical to show the results while still portraying the concepts well. We use the variance along an EPI line as a descriptor to measure the consistency with a particular region. Only the visible regions of each volume are taken into account to compute this variance. Thanks to the competition formulation, we use only two parameters in the segmentation. The first one μ regulates the smoothness of the extracted regions and second one in the initialization determines how close the estimated local directions need to be in order to be merged. This will in turn determine the number of regions to extract. The algorithm takes approximately 1000 seconds using the classical level set implementation to extract the coherent regions for both natural datasets analyzed. Improvements of several orders of magnitude are expected using more efficient implementations [16]. Figure 2 illustrates some of the images[2] and the final segmentations. Due to the absence of a ground truth, we provide a qualitative assessment. The

[1] We omit this proof due to lack of space.

[2] The authors would like acknowledge the Audiovisual Communications Laboratory at the Swiss Federal Institute of Technology (EPFL) for providing the equipment to capture these datasets.

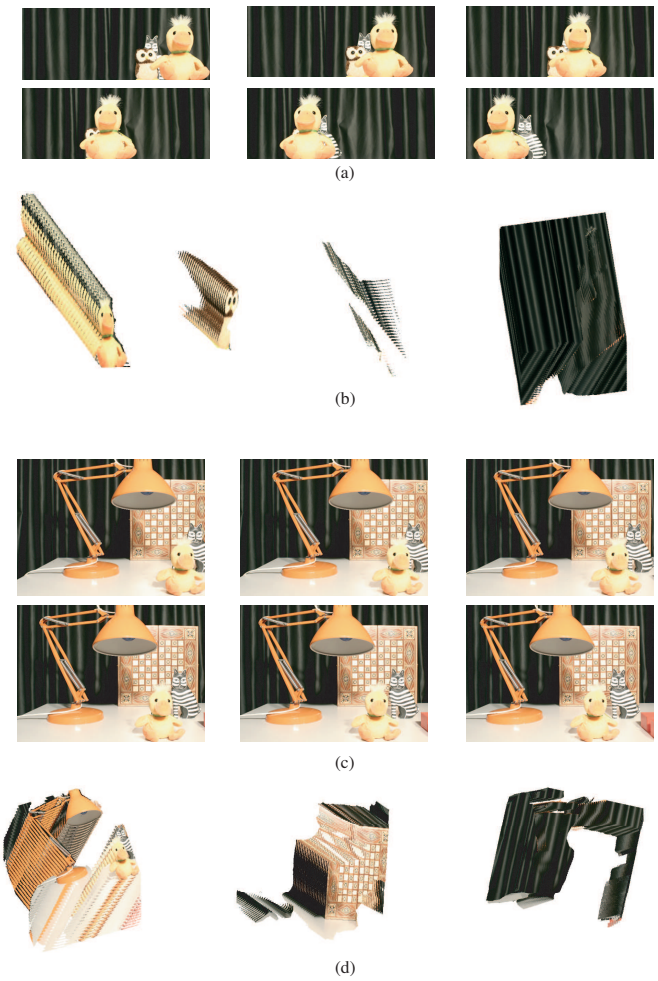


Figure 2: Automatically extracted coherent regions in simplified light fields. The first data set *animal family* consists of 32 images (235x625) some of which are illustrated in (a). The extracted regions at different depths are depicted in (b). The second data set has 25 images (400x500) some of which are illustrated in (c). The extracted volumes corresponding to the foreground, intermediate and background regions are depicted in (d).

regions are correctly captured despite the occlusions and disocclusions. The volumes also do not show the discontinuities across scan lines that may occur when individual EPIs (i.e. slices of the volume) are analyzed. All these aspects come from the fact that the data is analyzed in a global manner and coherence is enforced in all the dimensions. Some errors occur in large textureless regions due to the absence of a color based term in the energy minimization. These errors, however, will not significantly hinder the rendering quality.

4 Applications in image based rendering

The extraction of coherent regions in light fields is a useful step in numerous image based rendering applications. In this section, we demonstrate a few of these applications such as view interpolation and scene manipulations for object removal and insertion.

4.1 View interpolation

In the original light field rendering [13], the data is highly sampled and virtual viewpoints are rendered using classical quad-linear interpolation. This is equivalent to using a focal plane at infinite depth. In undersampled light fields however, this method causes blurring due to the aliasing that occurs. This sampling problem has been studied by Chai et al. [6] using a classical signal processing framework. The authors show that, due to the particular structure of the data, the band is approximately bound by the maximum and minimum depths in the scene. Moreover, it is shown that that there is an optimal rendering depth that will generate the new viewpoint with the least possible aliasing. However, the rate required often becomes very large when the scene has somewhat large depth variations. Furthermore, inherent effects such as occlusions are concentrated in space and hence contribute to spread the band of the function. Similarly to rendering with layers [15], the individual depth variations are much smaller in each of the extracted coherent volumes and an aliasing free rendering for each region can be achieved with less cameras. In order to demonstrate the reduced aliasing, we render a viewpoint using linear interpolation. Figures 3(a) and (c) illustrate rendered viewpoints using a single plane at the optimal depth in [6]. Note that some aliasing occurs in regions that are far from this depth. The images in Figures 3(b) and (d) illustrate the same rendered viewpoints generated by individually interpolating each of the extracted coherent regions. That is, the knowledge of occlusions is used and each of the regions are interpolated using their estimated constant depths. The aliasing is greatly reduced while keeping the natural aspect of the images.

4.2 Object manipulation

Similarly to layer based representations [21, 17], the extraction of the coherent regions enables to manipulate the data by recombining them in different ways. Note that a layer based representation (i.e alpha map, texture and plane or motion parameters) is not used as such. Rather, the new scenes are rendered using the full volumes. This enables to overcome some of the artifacts caused when over simplified depth models are used. The first example depicted in Figure 4 illustrates how the available coherent region $\mathscr{H}_n^{\perp}$ as shown in Figure 4(a) can be extrapolated to obtain the full region $\mathscr{H}_n$ shown in Figure 4(b). Linear extrapolation is performed using the available lines and their intensities. That is, each EPI line is simply extended along its slope.

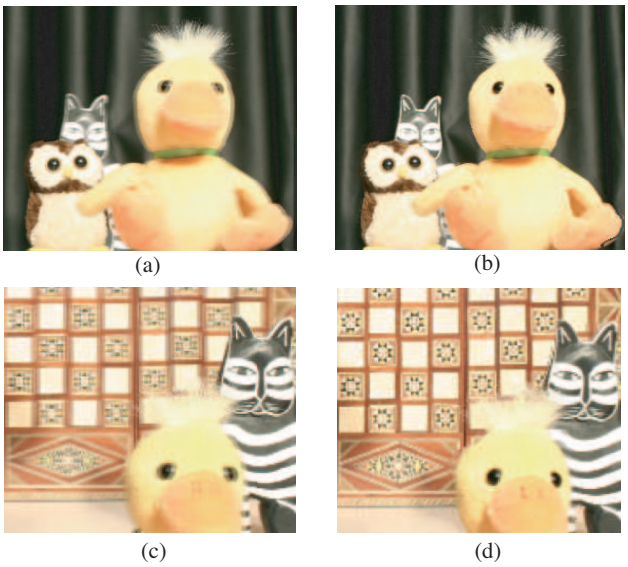


Figure 3: Image based rendering with coherent regions. The images in (a) and (c) show the rendered viewpoints using a single plane at the optimal depth. The images in (b) and (d) illustrate the same viewpoints rendered using the individual estimated constant depths for each of the extracted coherent regions.

Alternatively, the scene may be reconstructed by combining the existing coherent regions with external ones. These can be captured by other camera arrays or synthetically generated in order to perform augmented reality. As an example, we use a CAD software to generate the synthetic images of a teapot shown in Figure 5(a). The visible parts of the coherent region are determined using equation (3) which leads to the orthogonalized volume depicted in Figure 5(b). When combined with the original data, all the occlusions are naturally handled. Some of the rendered images are illustrated in Figure 5(c). Although the regions were extracted using fronto-parallel models, the objects still show their original shapes in the rendered images. This is noticeable, for instance, in the duck's beak and its shadow that are accurately rendered.

5 Conclusion and future work

In the first part of the paper, we proposed a novel multidimensional scheme based on a variational framework for the extraction of coherent regions in light fields. Since the formulation is global, coherence and consistency is enforced on all the dimensions. The method presented also takes fully advantage of the constraints namely that points in space are mapped onto particular trajectories (i.e. lines in light fields) and occlusions occur in specific order. Therefore occlusions are naturally handled and all the images are treated equally. The second part illustrated some applications in image based rendering using the extracted regions. View interpolation and scene manipulation as well as augmented reality were demonstrated in order to illustrate the benefits of extracting these regions. Since the scene is not represented as layers but as a combination of coherent hypervolumes,

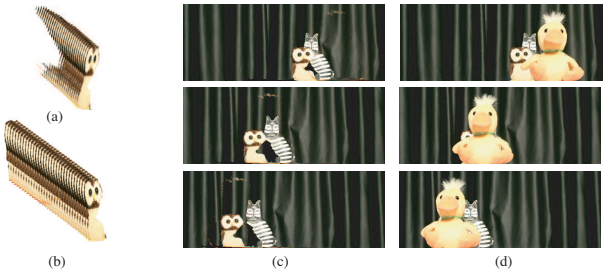


Figure 4: Coherent region extrapolation. The extracted volume illustrated in (a) is extrapolated along its EPI lines in order to reconstruct the occluded regions as shown in (b). This enables to reconstruct the scene, for example, by removing the region carved out by the duck and using the extrapolated background volumes (c). Note that there are some holes since some regions are never visible in the entire stack of images. The images in (d) show the original data for comparison.

accurate geometry is often not necessary and fronto-parallel depth models were used to extract the regions. More general depth models may also be incorporated into the energy minimization.

An interesting direction for future work is to explore the potential of the global formulation to automatically extract coherent regions in dynamic and unstructured light fields. This would imply using the constraints and imposing coherence in the four spatial dimensions as well as the time dimension.

References

[1] E. H. Adelson and J. R. Bergen. The plenoptic function and the elements of early vision. In M. Landy and J. Anthony Movshon, editors, *Computational Models of Visual Processing*, pages 3–20. MIT Press, Cambridge, MA, 1991.

[2] S. Baker, R. Szeliski, and P. Anandan. A layered approach to stereo reconstruction. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 434–441, 1998.

[3] J. Berent and P. L. Dragotti. Segmentation of Epipolar Plane Image volumes with occlusion and dissocclusion competition. In *IEEE Int. Workshop on Multimedia Signal Processing*, pages 182–185, October 2006.

[4] R.C. Bolles, H. H. Baker, and D.H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *Int. Journal of Computer Vision*, 1:7–55, 1987.

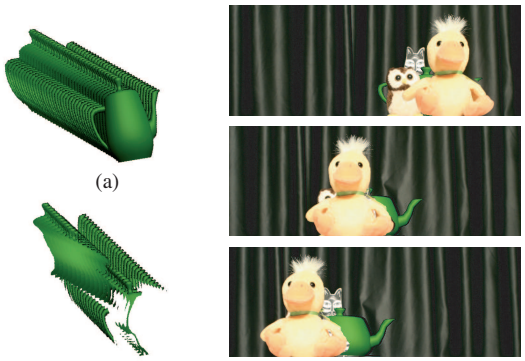
[5] C. Buehler, M. Boss, L. McMillan, S. Gortler, and M. Cohen. Unstructured lumigraph rendering. In *Computer graphics (SIGGRAPH '01)*, pages 425–432, 2001.

[6] J.-X. Chai, S.-C. Chan, H.-Y. Shum, and X. Tong. Plenoptic sampling. In *Computer graphics (SIGGRAPH '00)*, pages 307–318, 2000.

[7] S. C. Chan, K. T. Ng, Z. F. Gan, K. L. Chan, and H. Y. Shum. The plenoptic video. *IEEE Trans. on Circuits and Systems for Video Technology*, 15:1650–1659, 2005.

[8] A. Criminisi, S. B. Kang, R. Swaminathan, R. Szeliski, and P. Anandan. Extracting layers and analyzing their specular properties using epipolar-plane-image analysis. *Computer Vision and Image Understanding*, 97(1):51–85, January 2005.

[9] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In *Computer graphics (SIGGRAPH '96)*, pages 43–54, 1996.



(a) (b) (c)

Figure 5: Image based object insertion. The synthetically generated coherent region carved out by a teapot (a) is geometrically orthogonalized with the existing extracted volumes (b). It is then straightforward to recombine the coherent regions to recreate a scene where the teapot is inserted (c).

[10] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[11] S. Jehan-Besson, M. Barlaud, and G. Aubert. DREAMS: Deformable regions driven by an Eulerian accurate minimization method for image and video segmentation. *Int. Journal of Computer Vision*, 53(1):45–70, 2003.

[12] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int. Journal of Computer Vision*, 1(4):321–331, 1988.

[13] M. Levoy and P. Hanrahan. Light field rendering. In *Computer graphics (SIGGRAPH '96)*, pages 31–42, 1996.

[14] M. Lin and C. Tomasi. Surfaces with occlusions from layered stereo. In *IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 710–717, 2003.

[15] Z. Lin and H.-Y. Schum. A geometric analysis of light field rendering. *International journal of computer vision*, 58(2):121–138, 2004.

[16] J. Sethian. *Level Set Methods*. Cambridge University Press, 1996.

[17] J. Shade, S. Gortler, L. W. He, and R. Szeliski. Layered depth images. In *Computer graphics (SIGGRAPH '98)*, pages 231–242, 1998.

[18] H. Y. Shum, J. Sun, S. Yamazaki, Y. Li, and C. K. Tang. Pop-up light field: An interactive image-based modeling and rendering system. *ACM Trans. Graph.*, 23(2):143–162, April 2004.

[19] J.E. Solem and N.C. Overgaard. A geometric formulation of gradient descent for variational problems with moving surfaces. In *Int. Conf. on Scale Space and PDE methods in Computer Vision, Scale Space 2005*, 2005.

[20] T. Stich, A. Tevs, and M. Magnor. Global depth from epipolar volumes - a general framework for reconstructing non-lambertian surfaces. In *3DPVT*, pages 1–8, 2006.

[21] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *IEEE Trans. on Image Processing Special Issue: Image Sequence Compression*, 3(5):625–638, September 1994.

[22] C. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High-quality video view interpolation using a layered representation. In *Computer graphics (SIGGRAPH '04)*, pages 600–608, 2004.

Local image Features for Shoeprint Image Retrieval

H. Su, D. Crookes, A. Bouridane, and M. Gueham
School of EEE and CS, Queen's University Belfast
Email: {h.su, d.crookes, a.bouridane, mgueham02}@qub.ac.uk

Abstract

This paper deals with the retrieval of scene-of-crime (or scene) shoeprint images from a reference database of shoeprint images by using a new local feature detector and an improved local feature descriptor. Our approach is based on novel modifications and improvements of a few recent techniques in this area: (1) the scale adapted Harris detector, which is an extension to multi-scale domains of the Harris corner detector; (2) automatic scale selection by the characteristic scale of a local structure. (3) SIFT (Scale-Invariant Feature Transform), one of the most widely investigated descriptors in recent years. Like most of other local feature representations, the proposed approach can also be divided into two stages: (i) a set of distinctive local features are selected by first detecting scale adaptive Harris corners where each of them is associated with a scale factor, and then selecting as the final result only those corners whose scale matches the scale of blob-like structures around them. Here, the scale of a blob-like structure is detected by the Laplace-based scale selection. (ii). for each feature detected, an enhanced SIFT descriptor is computed to represent this feature. Our improvements lead two novel methods which we call the Modified Harris-Laplace (MHL) detector, and the enhanced SIFT descriptor.

In this paper, we demonstrate the application of the proposed scheme to the shoeprint image retrieval problem using six sets of synthetic scene images, 50 images for each, and a database of 500 reference shoeprint images. The retrieval performance of the proposed approach is significantly better, in terms of cumulative matching score, than the existing methods used in this application area, such as edge directional histogram, power spectral distribution, and pattern & topological spectra.

Key words: Shoeprint image, Forensic application, Local image features, Content-based image retrieval.

1 Introduction

Shoeprint Classification Shoeprints are often found at crime scenes and provide valuable forensic evidence. It has been estimated that more than 30% of all burglaries provide usable shoeprints that can be recovered from the crime scene [1]. Because of the pattern of repeated offences, rapid classification of such shoeprints would enable investigating officers not only to link different crimes, but to identify potential suspects while the crime is still 'hot'.

The objectives of the forensic investigator are usually: (i) to determine the make and model of a shoe; (ii) to determine if a particular shoeprint was made by a specific shoe; (iii) to match the shoeprint with other shoeprints, possibly from other crime scenes. An image of a shoeprint is first obtained using a technique such as photography, gel or electrostatic lifting, or by making a cast when the impression is in soil, snow or sand. A matching of the shoeprint is then made against specific image databases containing current and previous shoeprint images and/or images of shoeprints found at other crime scenes, leading to a few of candidate images, which are then used in the second stage. This matching process is usually in two stages: an initial classification which results in a hitlist of likely candidates, followed by a visual identification from this reduced set. For this paper, we are concerned with automating the first classification process.

The study of automatic shoeprint classification and retrieval is still relatively new and immature. Some work has recently been reported in [2-4, 28, 29]. Usually, these algorithms first automatically extract a set of feature vectors from a shoeprint image using different kinds of descriptors, such as Fourier descriptors, Fractals, edges, or pattern and topological spectra, and then sort a database of shoeprints in response to a sample shoeprint. However, several practical difficulties exist which hinder the effectiveness of real world shoeprint classification – both manual and automated, such as capture device-dependent noise, distortions, and cutting-out. In this paper, we generate six sets of synthetic scene images to simulate scene images, from a 'clean' dataset with 500 shoeprints by adding Gaussian noise or scene backgrounds, cutting-out, rescaling, rotating, or selecting salient patterns.

Local Feature Descriptors These are computed from distinctive local regions, and do not require segmentation beforehand. They have proved to be very successful in the applications such as image retrieval and matching [5-8], object recognition and classification [9-13], and wide baseline matching [14]. Consequently, many different scale- and affine-invariant local feature detectors, robust local feature descriptors, and their evaluations have been widely investigated in the literature [15-25].

Typically, a local image feature should have four properties: locality, repeatability, distinctiveness, and robustness to different degradations. The above studies suggest that no single local feature representation so far can beat others in terms of all the above four properties, so a good local feature representation should be a trade off of these properties. The work described in this paper firstly detects a set of distinctive local features from an image by combining a scale adaptive Harris corner detector with a Laplace-based scale selection. Then, for each local feature, an enhanced SIFT [18] descriptor is computed to represent the feature. Finally, the matching of descriptors in two images has been conducted by combining the nearest neighbour with the threshold-based screening, i.e. two descriptors are matched only if one is the nearest neighbour to the other one in the feature space, and at the same time the distance between them is less than a threshold. The distance between two images is computed from only the matched pairs.

The main contributions of this paper are: first, a new robust and distinctive local feature detector, Modified Harris-Laplace; second, an enhanced SIFT descriptor has been presented for the local features detected at the first stage; finally, we have investigated the application of these new local image features to the forensic application of automated shoeprint image retrieval, and compare this technique with other three up-to-date methods on the same sets of shoeprint images, and in terms of the same evaluation measure.

This paper is organized as follows: In section 2, we present the new local feature detector based on the adaptive Harris corner detector and the Laplace-based automatic scale selection. Section 3 introduces the enhanced SIFT descriptor. In section 4, we present the image databases used in this work, and also the experiments of shoeprint image retrieval based on local image features. We then summarize the paper in section 5.

2. Modified Harris-Laplace Detector

A local feature here refers to any point and its neighbourhood in an image where the signal changes significantly in terms of two dimensions. Conventional "corners", such as L-corners, T-junctions and Y-junctions satisfy this, but so do the isolated points, the endings of branches and any location with significant 2D texture. Further, all of these local structures have their characteristic size. K. Mikolajczyk et al in [6] have proposed a new Harris-Laplace detector by using the benefit from the Harris corner detector [26] (high accuracy of location) and that from LoG [27] (robust scale selection). But the way they combine the two powerful techniques does not necessarily result in an accurately located and stable scaled local feature detector, since this detector is actually to check if the response of the Harris measure reaches a maximum at the spatial domain and so does the response of the LoG at the same location along the scale direction. In most cases, the unstable component of this detector is the scale selection, since the stability of the scale selection based on LoG is conditioned that this measure should be computed at the centre of a blob structure. In this section, we shall present our solution to combine the two powerful techniques. Considering the Harris-Laplace detector in [6], we name our detector Modified Harris-Laplace.

The scale adaptive Harris detector [6] is based on an extension of the second moment matrix as shown Formula (1), where, σ_i is the integration scale, σ_d is the differentiation scale and f_α is the derivative computed in the direction of α. The strength of a scale adaptive corner can be measured by $\det(\boldsymbol{A}(\boldsymbol{x},\sigma_i,\sigma_d)) - \kappa \cdot \boldsymbol{trace}^2(\boldsymbol{A}(\boldsymbol{x},\sigma_i,\sigma_d))$.

$$\boldsymbol{A}(\boldsymbol{x},\sigma_i,\sigma_d) = \sigma_d^2 \cdot \boldsymbol{g}(\sigma_i) * \begin{bmatrix} \boldsymbol{f}_x^{\,2}(\boldsymbol{x},\sigma_d) & \boldsymbol{f}_x \boldsymbol{f}_y(\boldsymbol{x},\sigma_d) \\ \boldsymbol{f}_y \boldsymbol{f}_x(\boldsymbol{x},\sigma_d) & \boldsymbol{f}_y^{\,2}(\boldsymbol{x},\sigma_d) \end{bmatrix} \tag{1}$$

$$\boldsymbol{LoG}(\boldsymbol{x},\sigma) = \sigma^2 \left| \boldsymbol{L}_{xx}(\boldsymbol{x},\sigma) + \boldsymbol{L}_{yy}(\boldsymbol{x},\sigma) \right| \tag{2}$$

An example of the scale-adaptive Harris corner detection on a synthetic image is shown in Fig. 1 (a), where the figure suggests that for a corner, there might be a series of points detected by this approach due to the diffusion and the spatial extent of the corner. Obviously, only a few of them (normally one or two) present the characteristic size of this corner.

The normalized LoG response of an image is defined as Formula. (2), where σ is the scale, and $\boldsymbol{L}_{\alpha\alpha}$ is the second derivative computed in the direction of α. The LoG based automatic scale selection is based on the fact that the characteristic scale of a local structure very often corresponds to the local extremum of the LoG along the scale direction at the centre of a local blob (see the right curve of Fig. 1 (b)). But it must be noted that in most cases, this principle does not work well for the structures like corners. In Fig. 1 (b), the middle curve shows the LoG responses over scales, at the top-left corner of a square. The figure clearly illustrates why the scale selection of the Harris-Laplace detector [6] is unstable. There are too many extrema on the curve of the LoG responses, resulting in redundant and unstable corners.

However, Fig. 1 also suggests that the scale of the corners, located around a blob, might be related to the scale of that blob, which can be selected by referring to the LoG response. In real world, it is reasonable, in most cases, to assume that a corner can be associated with a blob structure around this corner, since any corner is just a part of an object. Based on this assumption, only those candidate corners, whose scale has a predefined relationship with the scale of a blob structure around them, can be selected as a characteristic scale of a corner. Here, two factors should be considered for this strategy: one is the searching region, the other one is the relationship between the scale of a blob structure and that of a corner. Fig. 2 illustrates this strategy, where the long-dashed circle ($\boldsymbol{r}$) denote a blob structure, and the short-dashed circles

denote the search region with the radius of r_1 and the reference circle with the radius of r_0. The solid circles are the candidates of the same corner located at the up-left of the square, and the bold solid circle is eventually selected as the characteristic scale of this corner. In all of our experiments, Formula (3) is applied to relate the reference scale r_0 and the search region r_1 with the blob scale r. (This formula is derived by assuming a square blob)

$$\sqrt{2} \cdot r_0 = r = \frac{\sqrt{2}}{2} \cdot r_1 \tag{3}$$

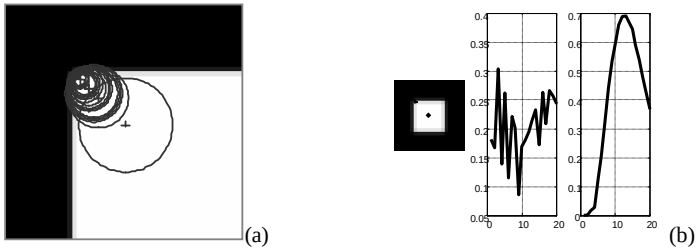


Figure 1 (a), an example of adaptive Harris corner detection, and LoG based scale selection. (b), the responses of the LoG measures at different locations along the scale direction (x-axis) on a synthetic image at the left. The middle curve corresponds to the top-left corner in the left image, and the right curve corresponds to the centre of the square in the same image.

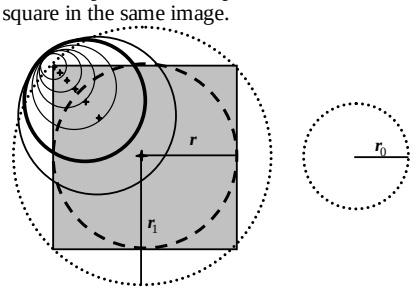


Figure 2 An illustration of automatic scale selection based on the scale of a blob-like structure

3. Local Feature Descriptors

The local photometric descriptors computed in this paper, as mentioned in the first section, are a further extension of the GLOH (Gradient Location and Orientation Histogram) [23, 25], originally from SIFT descriptors. Our method is different from GLOH in that:

(i) First, we apply a circular binary template on each normalized local region to increase the rotation invariance of the descriptor. This is a minor change but has a significant impact on the robustness of the descriptors. Either SIFT or GLOH obtains the rotation invariance by weighting the local region with a Gaussian window. However, it is very often the case that when one chooses a larger sigma of the Gaussian kernel, the descriptors computed on this region are distinctive but rotation sensitive; on the other hand, when one chooses a smaller sigma of the Gaussian kernel, the descriptors are rotation invariant but not distinctive. In most cases, it is hard to choose a proper sigma of the Gaussian window. Therefore we apply a binary template to model the region as a circular one, and meanwhile use a larger sigma of the Gaussian window to keep the distinctiveness of the descriptors.

(ii) Second, we bin the histogram with the orientation range of 180° rather than the original 360° for inverse contrast robustness. In the application of shoeprint image matching, it is very

often that two images contain the same print but with inverse contrast, i.e. in one image the grey level of the print itself is larger than that of the background, whereas in the other image it is otherwise. This difference can be overcome by binning the histogram with the orientation range of 180°, i.e. without considering the polarity.

The construction of our local descriptors is similar to GLOH, i.e. we apply to bin the gradients a log-polar location grid with three bins in the radial direction and four bins in the angular direction (the central grid does not apply angular binning), resulting in a 9 location grid. Considering the orientation range of 180°, 4 bins are applied in gradient orientation. Finally, the descriptors of an image are composed by a $\boldsymbol{N} \times 36$ matrix, where $\boldsymbol{N}$ is the number of the local features detected in the working image.

The similarity between two images depends on the matching strategy of the local features. For the sake of retrieval speed, we apply the nearest neighbour and thresholding jointly to compute the distance between two images, i.e. for each descriptor in one image, the nearest neighbour in another image is found as a potential match, then only those matches whose distance is below a threshold are selected as the final matches. The similarity of two images is computed from the summation of $\exp(-\boldsymbol{d})$, where $\boldsymbol{d}$ denotes the distances of the final matches. Of course, there are many other strategies for computing the similarity or matching score between two images. For instance, the nearest neighbour can be applied to obtain the initial matches, and then the RANSAC (Random Sample Consensus) method is used to fit the geometric transformation and reject the mismatches, here the geometric transformation can be a fundamental matrix.

4. Experiments

4.1 Datasets

Our experiments and evaluations for shoeprint image retrieval are conducted on a database of 500 reference images, termed dClean. To simulate the scene images, a few of degraded image sets, have been generated from this base dataset.

(a) dNoisy - Gaussian noise of five different levels is added to each shoeprint in the base dataset. The noise level (σ) varies from 10 to 50 with a step of 10, where the range of the grey levels is from 0 to 255.

(b) dRotate - We have rotated each shoeprint in the base dataset with five random orientations in the range of 0°-90°. The selection of this range, but not 0°-360°, is based on the fact that the algorithms developed in this paper are flip invariant both in horizontal and vertical directions

(c) dRescale - We have rescaled each shoeprint in the base dataset with five random scale ratios in the range of 0.35 to 0.65.

(d) dPartial - Five partial shoeprint images are generated for each shoeprint in the first dataset. The percentage of the partial shoeprint which remains varies from 40% to 95%.

(e) dScene - We simulate the scene images from the base dataset. For each image in the base dataset, we randomly select five scene images from F&F Ltd as the background, and put the “clean” shoeprint on each of the background scenes, resulting in 2500 scene images.

(f) dComplex - This dataset contains 50 shoeprint images with complex backgrounds, geometric transformations, significant “cut-out”, or their combinations.

4.2 Other methods

Edge Directional Histogram (EDH) - The authors in [29] claimed that the interior shape information of a shoeprint image was described in terms of its significant edges. Therefore, they applied a histogram of the edge direction as the signature of a shoeprint image. This method first extract the edges using a Canny edge detector, then a total of 72 bins are used to bin the edge directions quantized at 5° intervals. To obtain rotation invariance, they compute the 1-D FFT of the normalized edge direction histogram and take it as the final signature of the shoeprint image.

Power Spectral Distribution (PSD) [4] – this method considers the power spectral distribution as the signature of shoeprint image. To compute the PSD, one needs to firstly down-sample an input image, and secondly takes a 2-D DFT on the down-sampled image, and then the power spectral distribution is computed, finally, a step of masking is taken to obtain the signature. In the similarity computation, the PSD of a query image has to be rotated 30 times, with 1° for each time, and the similarity takes the largest one over the 30 times. In our experiments, the step of rotation is removed because first it is very time consuming to exhaustively try each rotation, and secondly assuming the rotation degree in the range of 30° is not suitable at all for many cases, finally, from the technical point of views any signature can obtain the rotation invariance by this brutal way.

Pattern & Topological Spectra (PTS) [28] - This method starts by (i) de-noising a shoeprint image by a partial differential equation (PDE) approach; (ii) forming a binary shoeprint image by thresholding using Kittler's thresholding approach; (iii) this binary image is then filtered by a series of morphological operations (open) with size-increasing structuring elements. The pattern spectrum and the topological spectrum are constructed by either measuring the area or the Euler number of the filtered images against the size of the structuring elements.

4.3 Performance evaluation

Cumulative Matching Characteristics (CMC) - This measure is suitable for the case where the reference database contains only one relevant record for the query record, and it was claimed in [4] to be able to answer the question "What is the probability of a match if one looks at the first ***n*** percent of the database records?" and it can be estimated by computing the proportion of times during trails of the searching when the relevant record appears in the first ***n*** percent of the sorted records in the database. The performance of a retrieval system is reflected as a graph which plots the probability of a match (also called cumulative matching score, vertical axis) against the percentage of the records reviewed (horizontal axis).

4.4 Experimental results

The following experiments are to compare the performance, in terms of CMC, of four signatures: i.e. Edge Directional Histogram (EDH), Power Spectral Distribution (PSD), Pattern & Topological Spectra (PTS), and the Local Image Feature (LIF). The experiments are conducted as follows: we first take a shoeprint image from the six degraded datasets as a query image (considering the computation, we randomly choose 50 images from each of degraded datasets (all 50 images from dComplex), therefore, 50 trials for each degraded dataset), and search against the dataset of dClean. For each dataset, we compute the CMC curve, and the results are shown in Fig. 3. Further to the above quantitative comparisons, a few of retrieval examples are shown in Fig. 4, and table 1 also lists the signature sizes of the compared methods. These results suggest that:

(i) For the degradations of Gaussian noising, cutting-out, and rescaling, the signatures of PSD and LIF can almost achieve the perfect results, and LIF can also reach the similar results for the degradations of rotation and scene background addition.

(ii) The performance of EDH and PTS is a little bit worse than that of PSD and LIF for the degradations of Gaussian noising, cutting-out, rescaling, and rotation, but both of them still are efficient approaches considering that the cost (signature size) of the two signatures is significantly less than the other two. Within both of them, PTS beats EDH in most cases, while the only exception is on the rescaled database. This matches the results achieved in [28]. Furthermore, it is a surprise that both PTS and EDH beat PSD on the dataset of dScene, even though they both have much less cost than PSD does.

(iii) The signature of LIF works very well for all kinds of degradations. It clearly outperforms other signatures on the dataset with the hardest complex degradations. However, it also costs much more than the signatures of EDH, and PTS.

5. Conclusion

In this paper, we have proposed a new method, called LIF, which combines two novel improvements of existing techniques: Harris-Laplace detector and SIFT descriptor. The new local feature detector, coined Modified Harris-Laplace detector, applies in a new way the scale-adaptive Harris corner detector to determine the candidates of local features, and the Laplace-based automatic scale selection strategy to pick the final local features. We further improve the widely used local feature descriptors – SIFT to be more robust to rotation and inverse contrast. In order to test the performance, we have generated six sets of synthetic scene shoeprint images, and then conducted a few of experiments on shoeprint image retrieval.

The experimental results indicate that, considering the cost of a signature, PTS is a good choice for minor degradations. However, for the real scene images, which are of very bad quality, the proposed signature of LIF based on the Modified Harris-Laplace detector and the enhanced SIFT descriptor is no doubt the best choice for archiving them.

Further issues to be investigated include: (i) further reduce the dimensions of a local feature descriptor; (ii) a fast and accurate matching strategy is necessary for a large shoeprint image database. The current matching based on the nearest neighbour and threholding is fast but not accurate enough, while a more accurate matching strategy based on RANSAC is time-consuming; (iii) there is continuing scope for more advanced local feature detectors and descriptors.

6. Acknowledgements

This work has been supported by EPSRC Grant No. EP/C008057/01. The authors would also like to thank Foster & Freeman Ltd. and Forensic Science Services for providing the shoeprint images used in the evaluation.

References

[1] G. Alexandre, “Computerized classification of the shoeprints of burglar’s soles”, Forensic Science International, Vol. 82, pp. 59-65, 1996.

[2] Z. Geradts and J. Keijzer, “The image data REBEZO for shoeprint with developments for automatic classification of shoe outsole designs”, Forensic Science International, Vol. 82, pp. 21-31, 1996.
[3] A.G. Alexander, A. Bouridane, D. Crookes, “Automatic classification and recognition of shoeprints,” Special issue of the information bulletin for shoeprint/toolmark examiners, Vol. 6. No. 1, pp. 91-104, Mar. 2000.
[4] P. D. Chazal, J. Flynn, and R.B. Reilly, “Automated processing of shoeprint images based on the Fourier transform for use in forensic science,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 3, pp. 341-350, Mar. 2005.
[5] C. Schmid and R. Mohr, “Local greyvalue invariants for image retrieval,” IEEE Trans. on PAMI, Vol 19-5, pp. 530-535, 1997.
[6] K. Mikolajczyk and C. Schmid, “Indexing based on scale invariant interest points” ICCV’01, pp. 525—531, 2001.
[7] V. Gouet and N. Boujemaa, “Object-based queries using colour points of interest,” IEEE Workshop on Content-Based Access of Image and Video Libraries (CVPR/CBAIVL) Hawai,USA, 2001
[8] L. Ledwich, and S. Williams, “Reduced SIFT Features for Image Retrieval and Indoor Localisation,” Australasian Conf. on Robotics and Automation, Canberra, 2004.
[9] D. Lowe, “Object Recognition from Local Scale-Invariant Features,” ICCV’99, pp.1150-1157. 1999.
[10] X. Ma and W. Eric L. Grimson, “Edge-Based Rich Representation for Vehicle Classification”, ICCV’05, pp. 1550-5499, 2005.
[11] K. Mikolajczyk, B. Leibe, and B. Schiele, “Local Features for Object Class Recognition,” ICCV’05, pp.1792-1799, 2005.
[12] B. Platel, E. Balmachnova, L.M.J. Florack, and B.M. ter Harr Romeny, “Top-Points as Interest Points for Image Matching”, ECCV’06, Part I, LNCS 3951, pp. 418–429, 2006.
[13] Ran Gal and Daniel Cohen-Or “Salient Geometric Features for Partial Shape Matching and Similarity,” ACM Transactions on Graphics, Vol. 25, No.1, pp.130-150, 2006.
[14] T. Tuytelaars and L. Van Gool. “Matching widely separated views based on affine invariant regions,” IJCV, Vol. 59, No. 1, pp. 61–85, 2004.
[15] K. Mikolajczyk and C. Schmid, “Scale and Affine invariant interest point detectors,” IJCV, Vol. 60, No. 1, pp.63-86, 2004.
[16] T. Kadir, A. Zisserman, and M. Brady, “An affine invariant salient region detector,” ECCV’04, pp. 404-416, 2004.
[17] G. Dorko and C Schmid, “Maximally stable local descriptor for scale selection”, ECCV’06, pp.504-516, 2006.
[18] D. Lowe, "Distinctive image features from scale-invariant keypoints", IJCV, Vol.60, No. 2, pp. 91-110, 2004.
[19] H. Bay, T. Tuytelaars, L.V. Gool, “SURF: Speed-Up Robust Features”, ECCV’06, pp.404-417, 2006.
[20] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust wide baseline stereo from maximally stable extremal regions,” BMVC, Vol. 1, pp. 384--393, 2002.
[21] C. Schmid, R. Mohr, and C. Bauckhage, “Evaluation of interest point detectors”, IJCV, Vol. 37, pp. 151 - 172, 2000.
[22] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, L. Gool, “a comparison of affine region detectors”, IJCV, Vol. 65, No.1-2, pp.43-72, 2005.
[23] K. Mikolajczyk, C. Schmid, "A performance evaluation of local descriptors," CVPR’03, Vol. 2, pp. 257-264, 2003.

[24] K. Mikolajczyk, C. Schmid, “A performance evaluation of local descriptors,” IEEE Trans. on PAMI 27(10):1615-1630. 2005.
[25] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust wide baseline stereo from maximally stable extremal regions,” BMVC, Vol. 1, pp. 384--393, 2002.
[26] C. Harris and M. Stephens, “A combined corner and edge detector,” Alvey Vision Conference, pp. 147–151. 1988.
[27] T. Lindeberg, “Feature Detection with Automatic Scale Selection”, IJCV, Vol. 30, No. 2, 1998.
[28] H. Su, D. Crookes, and A. Bouridane, “Shoeprint image retrieval by topology and patterns”, Queen’s University Belfast, Tech. Rep. May 2006
[29] L. Zhang, and N.M. Allinson, “Automatic Shoeprint Retrieval System for use in Forensic Investigations,” 5th Annual UK Workshop on Comp. Intelligence, 2005

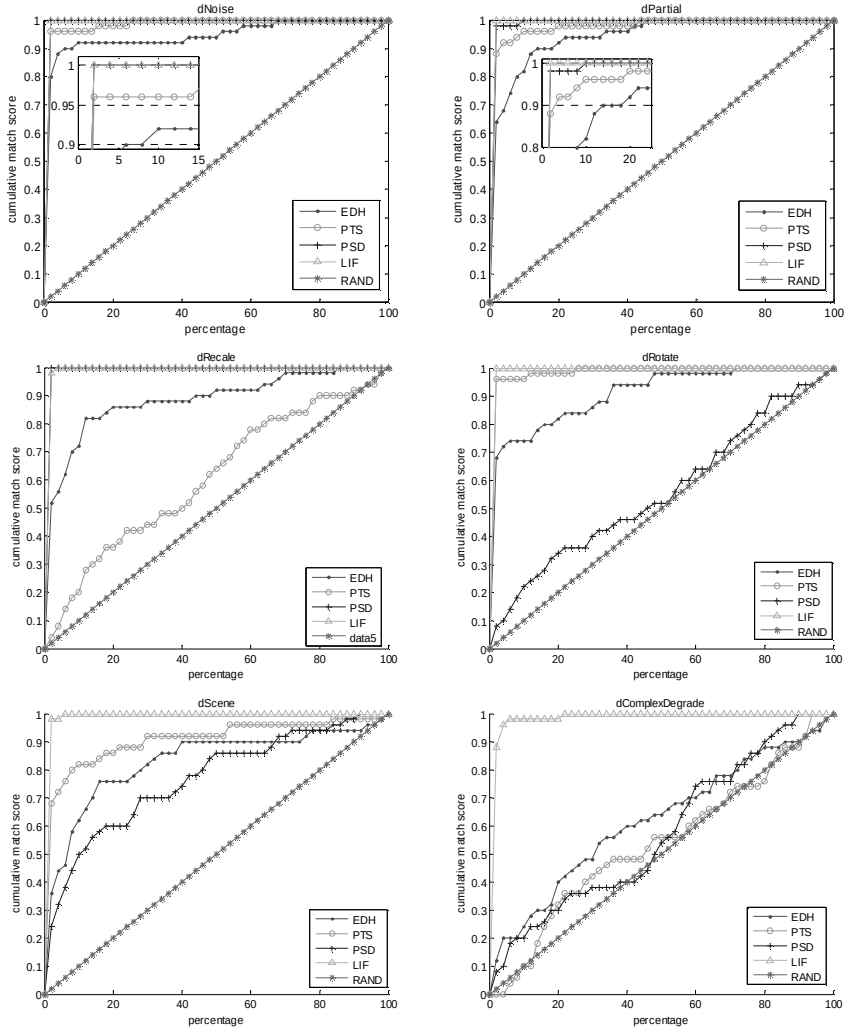


Figure 3 Performance evaluations of four signatures (EDH, PTS, PSD, and LIF) in terms of cumulative matching score on six degraded image dataset. RAND here is the worst case, i.e. the order of the images in the dataset is randomly allocated.

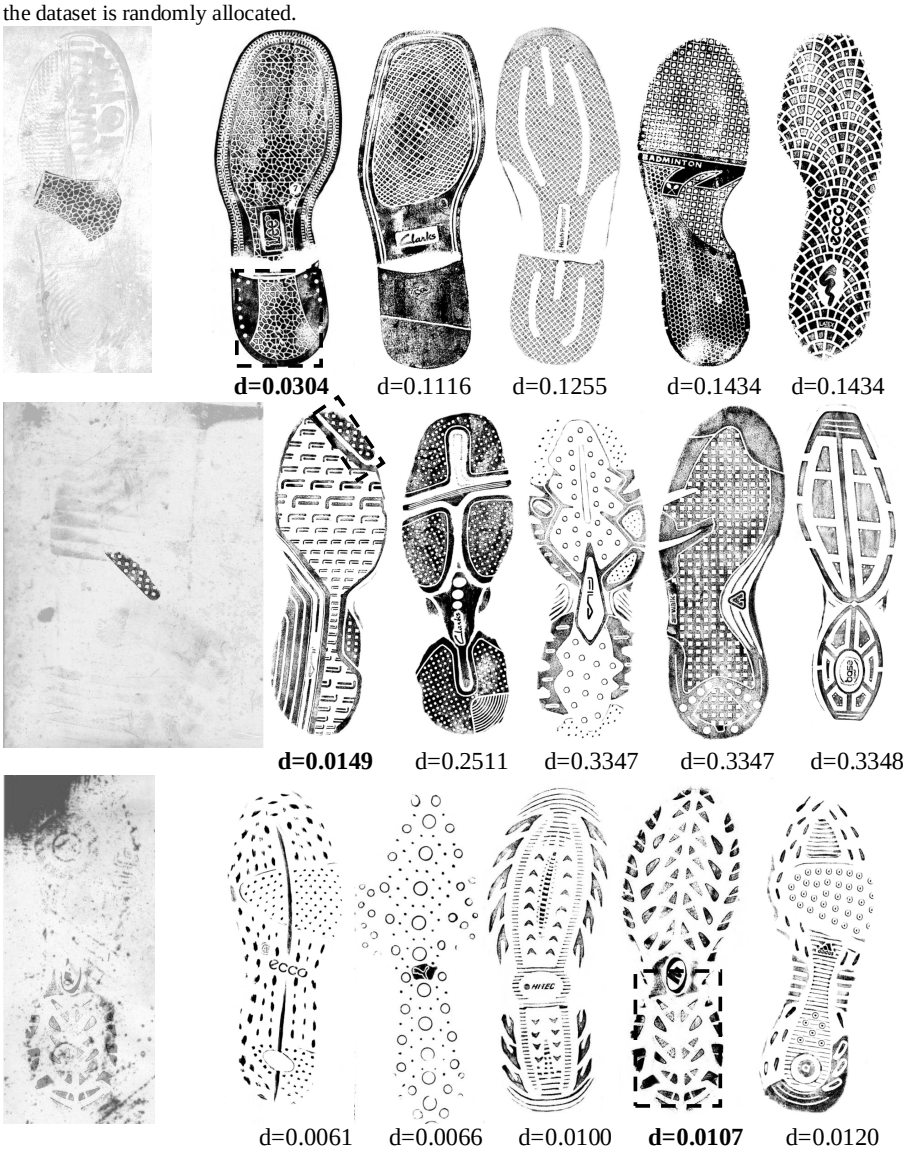


Figure 4 Examples of shoeprint image retrieval. In each row, the most left image is a noisy query shoeprint from dComplex, and the rest of the row is the top ranked shoeprint images in dClean. The distance is shown under each retrieved image, and the red squares denote the corresponding patterns contained in the query images.

Methods	EDH	PTS	PSD	LIF
Signature size	72	120	24,099	19,131

Table 1 the signature size of four techniques.

Colour Transfer by Feature Based Histogram Registration

C. R. Senanayake and D. C. Alexander
Department of Computer Science, UCL
Gower Street, London, WC1E 6BT
{c.senanayake|d.alexander}@cs.ucl.ac.uk

Abstract

This paper addresses the problem of automatically eliminating unwanted colour variation between similar image pairs. We propose a feature-based registration method to align colour histograms without any spatial processing of the image content or assumptions about the scene contents. We use the method for the colour transfer problem. The features are histogram maxima that persist through the scale space, so they are robust to large changes in the sizes of objects in the scene. The algorithm seeks the best matches between features and aligns the histograms via a polynomial warp. We construct a set of image pairs that exhibit variation in object scale and lighting and use it to show that the method produces better colour space alignments than simple alignments of histogram moments.

1 Introduction

A common problem in computer vision is that different sensors acquire different colour responses to an imaged object. This problem occurs because physical factors during the imaging process introduce a variation that differs for each sensor; in addition, it is practically impossible to image an object under perfectly constant lighting conditions at different spatial positions within an imaging environment [3][4][10]. This variation degrades the performance of colour computer vision processes such as object tracking [4]; in addition, the involved nature of calibration routines means that the calibration step is often ignored.

The colour transfer approach offers the potential for automatic alignment of similar colour spaces without manual intervention to perform the calibration. The goal of colour transfer methods is to make the image regions for the same object the same colour; in this paper, we seek to achieve this by aligning the corresponding dense regions of the histograms (clusters). We are only concerned with the situation when the images contain the same or at least highly similar objects. Existing colour transfer methods assume that scale changes between corresponding clusters are small; however, a significant change in object scale is common when the camera moves or when tracking between multiple cameras. We seek to develop a scale invariant cluster alignment algorithm and we pose colour transfer as a two-frame registration problem where our goal is to align the corresponding clusters in two colour histograms in the presence of lighting, automatic camera setting and object scale variation.

Registering colour histogram data presents unique challenges; corresponding clusters may change in shape and size, tear apart or merge in complex and uncharacterised ways under changes in lighting, camera variation and pose. The main contribution of this paper is to introduce a new method for automatically aligning histograms, that is robust to changes of scale of scene objects. The method matches scale invariant histogram features and aligns them using polynomial warps of the colour space.

2 Background

The term colour transfer was introduced to describe the process of aligning the colour histograms of a source and target image [8]. We distinguish colour transfer from the similar problems of colour consistency and colour constancy as follows: Colour consistency methods align colour histograms using correspondences established between objects of known reflectance such as calibration charts. Colour constancy methods estimate the light source and transform the colour histogram of an individual image to a standard reference frame using physical and/or statistical assumptions about the scene. Colour transfer on the other hand does not require objects of known reflectance in the scene and it computes an alignment using the histograms from both images.

The need to correct for sensor response and lighting variation occurs in a wide variety of settings such as 3D reconstruction [3], robotics [10] and special effects processing [8]. At the highest level, colour calibration methods divide into within camera and between camera calibrations. Within camera calibrations remove response variations that occur between different pixels in the same camera due to physical effects such as vignetting. These approaches involve prior physical models and manual calibration steps. Between camera calibrations align the response curves of different cameras. Five broad classes of between camera calibration methods exist in the literature, these are: chart based calibrations [3], calibrations using spatial overlap [9], disjoint view calibrations using non-parametric methods [7], disjoint view calibrations using parametric models [4] and calibrations using training images [10].

Polynomial transformations have been shown [3] highly effective in aligning colour histograms. However, no existing method can automatically establish the necessary correspondences to apply these transformations without colour calibration charts in the scenes. Our method overcomes this problem through a scale-invariant feature detection step that operates on the colour histograms. In addition, our approach involves no knowledge of the layout of objects in the scene and makes no parametric assumptions about the form of the colour data or the number of clusters that are present. FBHR has been developed using the RGB colour space. However, it contains no specific assumptions about the RGB space. We take this approach so the method can be applied to other colour spaces and applications in the future.

3 Method

The method we propose registers a source RGB histogram with a target RGB histogram. Registering the colour histograms of images computes a colour transformation that aligns the colours of a source image with those in a target image. Here, we only consider the case where the two images contain the same set of N single-coloured objects. Each colour

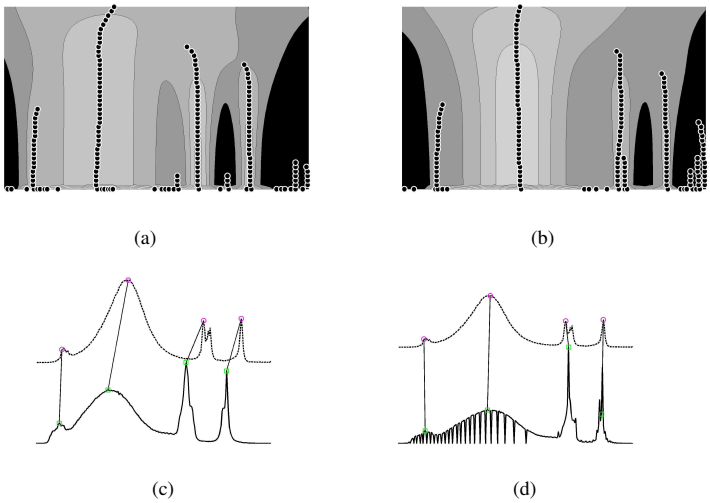


Figure 1: Example steps in the FBHR algorithm in the green channel when processing the BLSS image pair. 1(a) shows the scale space maxima for the green channel histogram of the source image (2(f)), 1(b) shows the corresponding scale space maxima for the green channel histogram of the target image (2(h)). (Circular dots indicate maxima, scale increases vertically and the intensity scale shows histogram density.) 1(c) shows a view of the detected features and the correspondences between the source histogram (solid line on bottom) and the target histogram (dashed line on top). 1(d) shows the warped source histogram and the target.
Histograms are plotted as lines joining the tops of the histogram bins, the transformed histogram is no longer smooth as it contains zero bins after being stretched by the multiplicative component of the transform.

histogram contains a number of dense regions that correspond to objects of interest. We seek the transform that aligns the clusters that correspond to the same object. The method is designed to handle multiple clusters of potentially different size, no explicit assumption is made about the shape of the distributions or the number of clusters present. We only assume that the source and target images are of the same set of objects.

3.1 Aligning Histogram Moments

Aligning histogram moments is an obvious candidate method for aligning colour distributions. Here, we outline three different moment transformations between a source histogram $\mathbf{S}$ and a target histogram $\mathbf{T}$. We compute an independent transformation in each dimension, we write the distribution in the ith channel as $\mathbf{S}_i$ and for the target distribution as $\mathbf{T}_i$: **1)** Additive alignment of the 1st moment: a scalar shift is computed for each channel $w_i = E(\mathbf{T}_i) - E(\mathbf{S}_i)$, where E is the mean operator; each scalar source data point s_i is transformed to $w_i + s_i$. **2)** Multiplicative alignment of the 1st moment: a scalar gain is computed from each channel $r_i = E(\mathbf{T}_i)/E(\mathbf{S}_i)$ and each source data point

is transformed to $r_i s_i$. **3)** Linear alignment of the 1st and second moment: the first and second moments can be aligned by a gain γ_i and a shift ω_i in each channel. γ_i is computed from the source histogram variance $V(\mathbf{S}_i)$ and target histogram variance $V(\mathbf{T}_i)$ such that $\gamma_i = V(\mathbf{T}_i)/V(\mathbf{S}_i)$. ω_i is computed as $\omega_i = E(\mathbf{T}_i) - \gamma_i E(\mathbf{S}_i)$, each source data point s_i is transformed to $\omega_i + \gamma_i s_i$.

3.2 Feature based histogram registration algorithm

In this section we outline the feature based histogram registration algorithm (FBHR). The steps to compute a colour transformation from a source image to a target image using FBHR are: **1)** Compute one dimensional histograms in each channel for both the source and target images. **2)** Compute the scale space of each histogram and extract salient features. **3)** Reject obvious outlying features. **4)** Match the remaining features. **5)** Compute the coefficients of polynomial transformations to align matching features in each channel. **6)** Transform the source image. **7)** Test for failure in each channel by comparing the transformed and target histogram. **8)** If FBHR fails, revert to a moment based transformation.

The following subsections elaborate on these steps and explain the rationale for taking this approach.

3.2.1 Scale Space features

We aim to match significant maxima in the colour histograms. Colour histogram data is commonly noisy and we avoid false feature generation using scale space smoothing. We track maxima that persist over a scale space. Analysis of the scale space in this way has been termed deep structure analysis [5] . The scale space of a function h can be computed by convolving h with Gaussian kernels $G(.;\sigma)$ of various widths σ. We detect all local maxima at each scale and remove the maxima that persist over less than T scales. We search from low levels of σ upwards. A feature persists between adjacent scales if its position is in adjacent histogram bins at the two scales; feature persistence is affected by histogram bin width and the values of σ. Figure 1 illustrates these features for one example image pair and their subsequent processing.

3.2.2 Rejecting outlier features and Matching

Oversaturation is a very common feature in digital images and leads to spikes at the ends of the histograms in question; we reject these features by thresholding out features less than δ away from the end of the range. Maxima found in histogram bins below a noise floor level γ are rejected as they are unlikely to be reliable.

The matching step seeks a match between the detected source and target histogram feature points. We do this with the following steps: **1)** Generate all matches between the source and target points for this number of matches. If there are the same number of source and target points we attempt to match them all, if either set is smaller we seek to match all the points in the smaller set. **2)** Reject matches that do not preserve rank ordering. In 1D this means that for a set of matched points, both points in each match must be either less than or greater than all other matched pairs. This prevents folding in the transformation. **3)** Pick the match with the minimum global Euclidean cost between points.

3.2.3 Point alignment transforms

The matching step outputs M_i corresponding points $(s_{ij}, t_{ij}), j = 1, ..., M_i$, in each channel $i = 1, ..., N$. We seek the transformation f_i in each channel that minimizes

$$\sum_{j=1}^{M_i} \left| f_i(s_{ij}) - t_{ij} \right|, \tag{1}$$

where $|.|$ is the L2 norm. We use an order d polynomial mapping for f_i so that

$$f_i(s_{ij}) = \alpha_{0i} + \sum_{k=1}^{d} \alpha_{ki} {s_{ij}}^k. \tag{2}$$

Each corresponding pair provides a linear constraint on the α_{ki}. We write them all as a matrix equation

$$\begin{bmatrix} 1 & s_{1,n} & s_{1,n}^2 & \dots & s_{1,n}^d \\ 1 & \vdots & \vdots & \vdots & \vdots \\ 1 & s_{M,n} & s_{M,n}^2 & \dots & s_{M,n}^d \end{bmatrix} \begin{bmatrix} \alpha_{0,n} \\ \vdots \\ \alpha_{d,n} \end{bmatrix} = \begin{bmatrix} t_{0,n} \\ \vdots \\ t_{d,n} \end{bmatrix}. \tag{3}$$

Writing this as $\mathbf{A}_i \mathbf{C}_i = \mathbf{D}_i$, we solve for $\mathbf{C}_i$: $\mathbf{C}_i = \mathbf{A}_i^\dagger \mathbf{D}_i$, where $\mathbf{A}_i^\dagger$ is the pseudo-inverse of $\mathbf{A}_i$. This is repeated for each channel.

3.2.4 Failure test

The added flexibility of the polynomial transformation makes them sensitive to mismatches and means that histograms can be poorly aligned when the correspondence step has not returned the correct result. We perform a coarse level of failure testing between the transformed histogram and target histogram in each colour channel. However, when the procedure fails, it tends to fail catastrophically, which is easy to detect. We compute the histograms for the transformed and target data less than δ away from the end of the range, this rejects over saturated regions which cause outlier effects. Then we compute the Bhattachayya coefficient $B_i = \sum_{b=1}^{P} \sqrt{r_b t_b}$ in each channel, where b indexes the transformed histogram bins r_b and the target histogram bins t_b. If B_i is less than a failure threshold F then FBHR has failed in this channel.

4 Experiments and Results

The experiments test the hypothesis that the feature based histogram registration method in section 3.2 transforms the colours of the source image histogram so that they are better aligned with the target histogram than when using alignment of either the 1st, or 1st and 2nd histogram moments (as in section 3.1). We investigate the failure points of the algorithm and simple strategies for mitigating these failures. The test images divide into three sets: 1) images of three distinctly coloured pieces of paper on a desk, 2) images

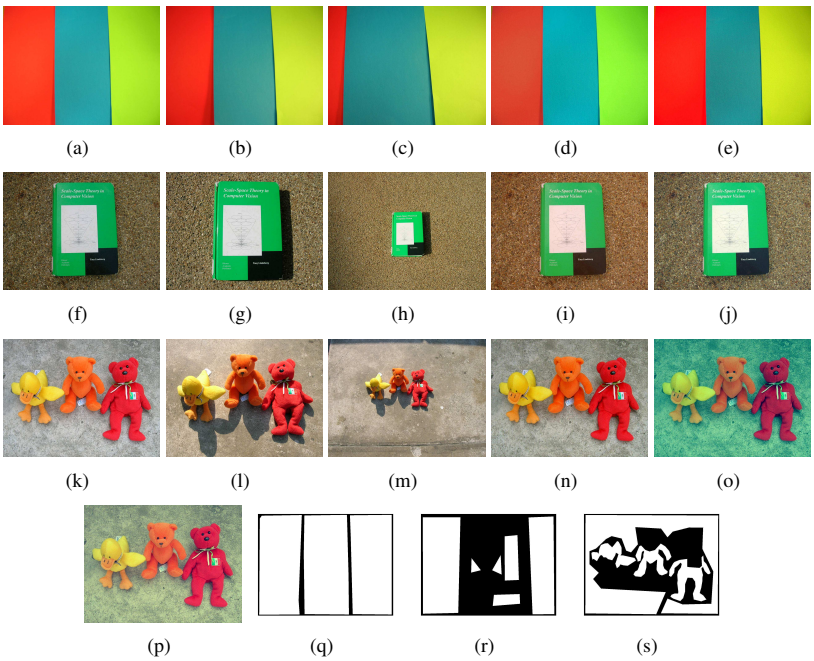


Figure 2: Source and target images from the three datasets, examples of transformed source images and masked regions. Image 2(a) is of paper strips under florescent lighting, 2(b) is taken under similar conditions but with a desk lamp turned on to cause a slight change in colours. Image 2(c) shows rearranged strips with the desk lamp on. Transformed source images are shown for a 1st moment shift 2(d) and FBHR1 2(e) when using source image 2(a) and target image 2(c) (SLSS). Image 2(f) shows the book at large scale in shadow and 2(g) shows it in sunlight, 2(h) shows the book at small scale in sunlight. Transformed images are shown for a 1st moment shift 2(i) and FBHR1 2(j) using source image 2(f) and target image 2(h) (BLSS). Image 2(k) shows the stuffed animals at large scale in the shade, 2(l) shows them in direct sunlight and 2(m) shows them at small scale and in sunlight. Transformed images are shown for a 1st moment 2(n), FBHR1 2(o) and FBHR1 shift fail 2(p) when using source image 2(k) and target image 2(l) (AL). Image 2(q), shows masked regions for 2(a), mask 2(r) shows the regions for 2(f) and 2(s) shows the mask for 2(k); white polygons indicate the regions of interest, black regions are ignored in the mask. For each mask, polygons corresponding to different colour regions of interest can be identified separately.

Table 1: Descriptions of image pairs used and short hand notation to refer to them. (Where possible, we reference the source (Src) and target (Tgt) images in Figure 2.

Code	Image pair description
SL	Paper strips under lighting change at same scale. (Src: 2(a), Tgt: 2(b))
SLS	Paper strips under lighting change and moderate scale change.
SLSS	Paper strips under light change and large scale change. (Src: 2(a), Tgt: 2(c))
BL	Book under lighting change at same scale. (Src: 2(f), Tgt: 2(g))
BS	Book under the same lighting with moderate scale change.
BLS	Book under lighting change and moderate scale change.
BLSS	Book under lighting change and large scale change. (Src: 2(f), Tgt: 2(h))
BLSSS	Book under lighting change and very large scale change
AL	Stuffed animals under a lighting change (Src: 2(k), Tgt: 2(l))
AL2	Stuffed animals occupying small percentage of image under a lighting change
ALSL	Stuffed animals under a lighting change and scale change

of a book and 3) images of stuffed toys. The image sets are constructed to pick source and target images that exhibit either scale, lighting or both scale and lighting changes. In the images of stuffed animals, shadows are a significant source of variation that make the alignment task more difficult. Figure 2 illustrates the types of images present in the image sets and the associated hand marked masks used in the evaluation of the colour transfer results. Table 1 describes the source and target image pairs and introduces a shorthand used to refer to image pairs from this point onwards.

We compare the alignment of moments to the FBHR method using polynomials of order 1, 2 and 3. The scale space was generated over T scales with $\sigma_n = e^{0.1(n-1)}$, $n = 1, \cdots, T$. The exponential increase in kernel scale ensures that blurring occurs more slowly at lower scales which prevents premature breakages in the feature paths that we are tracking. In all cases, we set the over-saturation threshold δ to 3, the noise-floor threshold γ to 0.001, T to 15 and the Bhattachayya failure threshold F to 0.7. The histogram bin width is 1 in all cases.

After alignment, we compute a metric that compares the colours of L corresponding regions marked by hand polygons in all images. The paper strip set is marked up into three regions, the book and stuffed animals each have four marked up regions; examples of marked up regions are given in figures 2(q), 2(r) and 2(s). The metric is the total average Mahalanobis distance between the corresponding regions in the transformed and target images. We denote the RGB pixels extracted from the nth masked region of the transformed image as a 3 column matrix $\mathbf{Q}_n$ and from the target image as a 3 column matrix $\mathbf{W}_n$ where each row in both $\mathbf{Q}_n$ and $\mathbf{W}_n$ corresponds to a RGB pixel extracted from the masked region associated with the image, we compute the Mahalanobis distance in each direction as

$$\phi_n = \sqrt{(E(\mathbf{Q}_n) - E(\mathbf{W}_n))^{-1} \mathbf{C}_{\mathbf{Q}_n} E(\mathbf{Q}_n) - E(\mathbf{W}_n)} \tag{4}$$

and

$$\beta_n = \sqrt{(E(\mathbf{W}_n) - E(\mathbf{Q}_n))^{-1} \mathbf{C}_{\mathbf{W}_n} E(\mathbf{W}_n) - E(\mathbf{Q}_n)}, \tag{5}$$

where $E(\mathbf{Q}_n)$ and $E(\mathbf{W}_n)$ are the mean RGB colours of $\mathbf{Q}_n$ and $\mathbf{W}_n$; $\mathbf{C}_{\mathbf{Q}_n}$ and $\mathbf{C}_{\mathbf{W}_n}$ are the covariance of $\mathbf{Q}_n$ and $\mathbf{W}_n$ respectively. For each pair of corresponding regions we find the

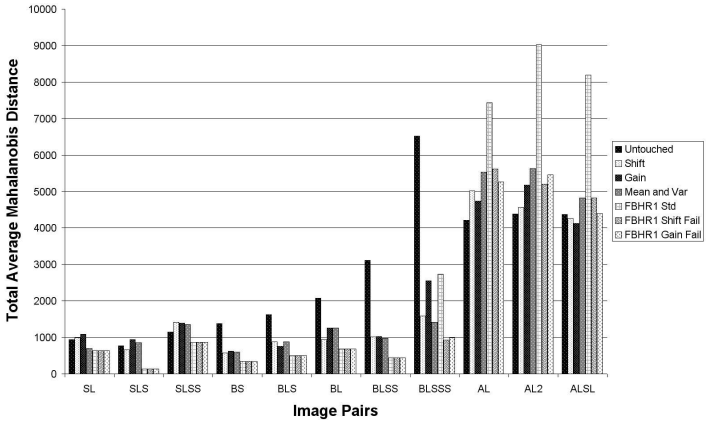


Figure 3: Total average Mahalanobis distances for different image pairs aligned using alignment of moments and FBHR1 and FBHR1 with different failure strategies. FBHR1 shift first applies FBHR1 and then applied a shift alignment of the 1st moment in any channel that fails. FBHR1 applied a gain shift of the 1st moment in any channel that fails.

average of these distances

$$J_n = \frac{\phi_n + \beta_n}{2}, \tag{6}$$

and the overall metric is the sum of the averages of ϕ_n and β_n, $\sum_{n=1}^{L} J_n$. The metric provides us with an overall measure of how closely the corresponding colour clusters are aligned, the measure is chosen over standard overlap measures such as the Bhattachayya coefficient as all clusters do not necessarily overlap after alignment. Overlap measures are dominated by largest overlapping clusters; however, we are interested in how well all the different colours in the scene are aligned. This metric is sensitive to separations between the means of corresponding clusters, but is potentially insensitive to orientation differences between clusters with closely aligned means. Manual inspection suggests that such configurations rarely arise, but we remain wary of this limitation.

Figure 3 summarises the result of applying the different moment alignment strategies and FBHR with two different failure strategies on all different datasets. The FBHR Shift and Gain fail transforms first attempt to apply FBHR in each channel, if a failure is detected they revert to alignment using a shift or gain alignment of the first moments respectively. We see that for the datasets SL,SLS,SLSS,BS,BLS,BL and BLSS no FBHR channel failure occurs and FBHR outperforms alignment using moments. On the BLSSS example the channel failure methods improve over FBHR1 as a channel has failed, this strategy also outperforms aligning moments. For the stuffed animal image pairs, AL and AL2 show worse alignment scores under all transformations and ALSL shows slight improvement when aligning moments. On these images the significant shadowing effects cause cluster variation that cannot be resolved by examining the individual R,G and B histograms alone. FBHR failure modes cause the standard FBHR algorithm to degrade in performance more gracefully. However, only the BLSSS pair shows an improvement over

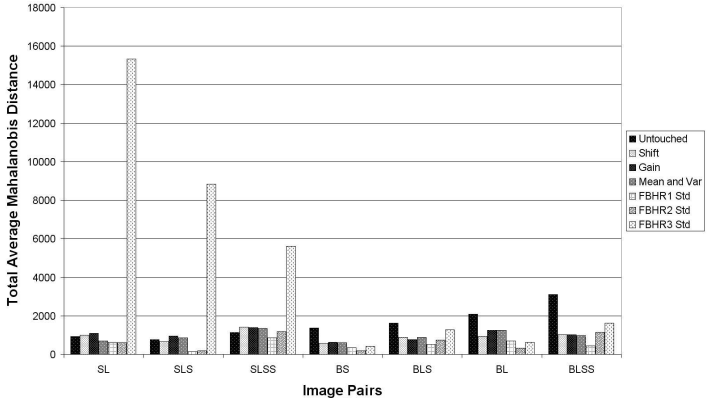


Figure 4: Comparison of moment alignment and FBHR for 1st, 2nd and 3rd order warps for image pair examples that did not detect FBHR failure in any channel.

moment alignment when using this strategy. This tells us that a practical strategy when FBHR failure has been detected is to align the 1st moments of all channels using a shift or gain; applying moment alignment transformations to individual channels on failure does not provide a significant overall improvement.

A further question is how the performance of FBHR compares with polynomial transformations of order 1, 2 and 3. Figure 4 examines FBHR 1, 2 and 3 for the example datasets that did not cause FBHR failure in a channel, we see that FBHR1 outperforms alignment of moments across all examples. Either FBHR1 or FBHR2 are found to perform best. Order 3 fails catastrophically for the strips data set because the histograms have too few features to fit an order 3 polynomial. The book images provide enough features, but the regularizing effect of the lower-order transformations proves advantageous. Order 2 provides little improvement over order 1 so we recommend order 1 by Occam's razor.

The method cannot automatically select the best transform which must be done empirically. However, FBHR 1 has proven robust across a wide range of examples. Note that alignment of the 1st and 2nd moment and FBHR 1 are both linear transformations so results are directly comparable. The fact that FBHR1 produces better alignments validates the feature based registration approach over moment alignment.

5 Conclusions and further work

We have introduced a new algorithm capable of aligning colour histograms that are noisy and include variation due to lighting, scale and small shadow changes (as seen in the book dataset). We have tested the algorithm for aligning a range of colour histograms obtained from pairs of images with systematically introduced variation of lighting, scale and shadowing. The experiments align the 1D individual R, G and B distributions of source and target distributions and have shown quantitative improvements in the 3D RGB colour histogram alignment over the simple methods of aligning moments. We have shown the merits of a feature-based histogram registration approach for aligning multi-modal data

and how to deal with its limitations.

One potential improvement would compute features directly from the 3D histograms rather than the 1D histograms of each colour channel. However, peak detection in 3D is more difficult than in 1D. We also plan to test the algorithm on more complicated scenes and evaluate the performance of standard registration methods such as ICP; RANSAC will be explored as an outlier rejection method. An increased number of colours in the scene will increase the complexity of the matching, we plan to reformulate the matching procedure as a bi-partite matching problem that penalises folding correspondences. The Hungarian algorithm [1] is a candidate method for solving the matching problem in $O(n^3)$ time. We have concentrated on the two-frame histogram registration problem, but plan to extend the method to allow multiple-frame histogram registration; methods to align multiple 3D data scans [2] may provide insight into the best way to do this. Finally, we intend to evaluate whether recently introduced methods for histogram comparison across overlapping and non overlapping clusters such as [6] improve the experimental evaluation.

References

[1] Wikipedia-Hungarian algorithm description. http://en.wikipedia.org/wiki/hungarian_algorithm.

[2] D.W. Eggert, A.W. Fitzgibbon, and R.B. Fisher. Simultaneous registration of multiple range views for use in reverse engineering of CAD models. *Computer Vision Image Understanding*, 69(3):253–272, 1998.

[3] A. Ilie and G. Welch. Ensuring color consistency across multiple cameras. In *The 10th IEEE Conference on Computer Vision (ICCV)*, Beijing, China, October 2005.

[4] K. Jeong and C. Jaynes. Object matching in disjoint cameras using a color transfer approach. *Special Issue of Machine Vision and Applications Journal*, 2006.

[5] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Springer, 1994.

[6] H. Ling and K. Okada. Diffusion Distance for Histogram Comparison. *CVPR*, 1:246–253, 2006.

[7] F. Pitie, A.C. Kokaram, and R. Dahyot. N-Dimensional Probability Density Function Transfer and its Application to Colour Transfer. *ICCV'05 Volume 2*, pages 1434–1439, 2005.

[8] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley. Color transfer between images. *Computer Graphics and Applications, IEEE*, 21(5):34–41, 2001.

[9] G.Y Tian, D. Gledhill, and D. Taylor. Colour correction for panoramic imaging. *Sixth International Conference on Information Visualisation*, pages 483–488, 2002.

[10] L.A. Torres-Mendez and G. Dudek. Color Correction of Underwater Images for Aquatic Robot Inspection. *Lecture notes in computer science*, 3757:60, 2005.

Super-resolution of faces using the epipolar constraint

Richard J.M. den Hollander, Dirk-Jan J. de Lange and Klamer Schutte

TNO Defence, Security and Safety
Oude Waalsdorperweg 63
P.O. Box 96864, 2509 JG Den Haag, The Netherlands
richard.denhollander@tno.nl

Abstract

In this paper we present a super-resolution scheme specifically designed for faces. First, a face detector is used to find faces in a video frame, after which an optical flow algorithm is applied to track feature points on the faces. Given the set of flow vectors corresponding to a single face, we propose to use the epipolar geometry for rejecting outlying flow vectors. This will improve the registration of the face over multiple frames, and thus lead to an improved super-resolution image. An iterative backprojection method is used for acquiring the super-resolution images.

1 Introduction

Nowadays many public areas are secured by surveillance cameras. The quality of the recorded material is sometimes low, making reliable recognition of individuals difficult. The use of super-resolution (SR) techniques may be promising for improving the quality of such video material.

A few approaches to SR on faces have been reported in literature. In [4] a SR frame is computed using information from past and future low-resolution (LR) frames. It calculates optical flow between interpolated versions of these frames and the initial SR frame. The updated SR frame is the average of the current SR frame and warped versions of neighboring interpolated LR frames. The algorithm considers only 5 neighboring frames, in which not much movement of the face is to be expected. In [7] the motion between frames is also estimated using optical flow. A probabilistic scheme is employed which determines for each input frame whether the pixels are visible in the SR frame or not. Only the visible pixels are used to update the SR frame. A somewhat different type of approach is the use of prior face knowledge to build SR images. It works, however, with the requirement that the observed face has the same pose as the model, e.g. a frontal view [3]. An extension to multiple poses is possible, but the results are not yet convincing [11].

Thus far, SR has mainly been applied for rigid planar objects or scenes, due to the fact that alignment is relatively easy in this case. When the object or scene is not planar, i.e. for a full 3D object, the transformation will not be a simple one-to-one mapping of pixel positions. For instance, parts of the object may be occluded after it is rotated. Not every object point will therefore be visible in all the frames which show the object.

In general, the relation between different views of a rigid 3D object is governed by the epipolar geometry [9]. Faces, however, are non-rigid 3D objects since they may change expression. Though, in practice, facial expressions will not change constantly, so that the epipolar geometry can be assumed to hold approximately.

In this paper, we demonstrate a SR method for faces based on optical flow, which makes use of the epipolar geometry. Once the epipolar geometry is known, outliers among the flow vectors can be removed. With the resulting improved motion estimates, the SR face images will be improved as well.

In Section 2 the SR scheme for faces is introduced. Experimental results for the SR scheme are given in Section 3. Finally, in Section 4, we conclude the paper.

2 A super-resolution scheme for faces

The approach we follow for SR on faces is in line of [4, 7], where optical flow is used to determine the motion. However, optical flow is only computed for feature points inside the face regions of the frame, which are found by the use of a face detector. This makes it possible to compute for each face the epipolar geometry, which describes the change in face pose w.r.t. the camera. The reason for the restriction to faces, is that the applied optical flow algorithm uses a hierarchical search method which becomes too complex for the processing of all image pixels. For every face a number of feature points is selected which will be tracked by the optical flow algorithm. The flow vectors are then used to compute the epipolar geometry describing the change in face pose w.r.t. the camera. Outliers among the flow vectors are then removed by retaining only those vectors which support the epipolar geometry. The computed optical flow is not dense, and therefore the per-pixel flow vectors are found by averaging sets of neighboring feature flow vectors. The applied SR algorithm is based on backprojection of the difference between the current frame and the simulated LR frame. An overview of the scheme is shown in Fig. 1.

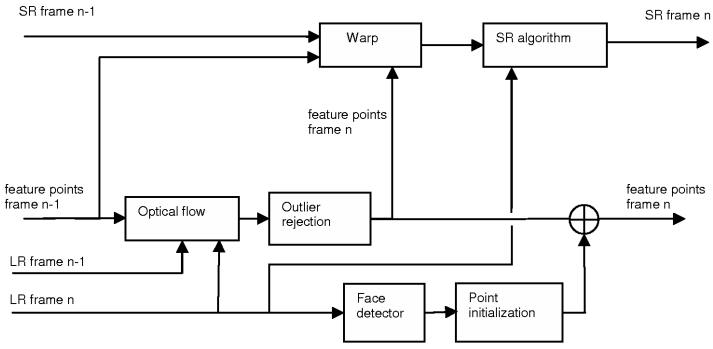
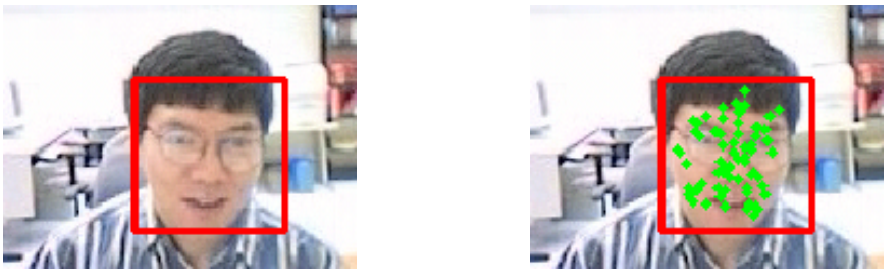


Figure 1: A schematic overview of the SR method for faces.

Next, we describe the separate elements in the SR scheme in more detail. For every element it is indicated whether it works on the whole frame (i.e. no subdivision per face) or on a per face basis.

2.1 Face detection - per frame

The face detector is taken from the OpenCV library [2], and is based on [12]. It is a fast detector which comes equipped with a set of trained classifiers. We have selected the default frontal face detector for our purpose, since recognition is best possible from frontal face views, and let it detect faces of size 20x20 pixels or larger. The classifier uses a collection of Haar features, which describe the differences in pixel sums between connecting regions. The features are examined in a cascaded classifier, which passes candidate face regions in a sequential fashion. When a region is not a face, it will be immediately rejected so that valuable processing time is saved. The output of the detector is a set of rectangles enclosing the detected faces, see Fig. 2(a). Note that although the images are displayed here in color, all processing is done on gray value images.



(a) (b)

Figure 2: A detected face in a video frame (a), and initialized feature points inside the rectangle (b).

2.2 Point initialization - per face

To estimate the motion of the face, a set of feature points (80 in total) is initialized inside the rectangle of the face detector. This is done because the hierarchical flow algorithm is too complex to apply for a dense pixel grid. The point set is connected to a single face and is processed separately in the **outlier rejection** stage. The points are equally spread inside a circle contained in the rectangle, which gives the highest probability of having all points on the face since the rectangle will typically cover parts of the background (see Fig. 2(b)). The circle radius is chosen such that it completely fits inside the rectangle.

Another option would be to find certain interest points in the image, like corners, and use these to initialize the points. These points may be more stable, but their positions in the face may not be equally spread. Since the displacement of pixels will depend on neighboring feature points (see **Warp** section), it is important to have sufficient features in each pixel's neighborhood.

2.3 Optical flow - per frame

The feature points of all faces are being tracked across the frames using a variant of the Lucas-Kanade optical flow algorithm from the OpenCV library [2]. This algorithm computes the flow using a pyramidal representation of the images [5]. It is set to run at

maximum 20 iterations or stop when the update is smaller than 0.03 pixel. In case the flow for a particular feature point can not be found, the point will be removed. Starting at the coarsest pyramid level and descending to the original size, the flow is computed at each level by using the flow of the previous level as initialization. The pyramid levels are created iteratively by low-pass filtering and factor 2 subsampling of the original image. Points falling outside the image borders will be marked as lost. When the displacement of the point is smaller than a threshold (0.2 pixel) in both x and y-direction, we assume that the point is locked on the static background and remove it. This is motivated by the fact that faces are expected to be continuously moving in the scene, that is, at least over one or more pixels per frame. In case there are no points left after the flow computation, the face will not be processed any further (except when it is detected again).

2.4 Outlier rejection - per face

As stated earlier, the displacements of correctly tracked points should obey the epipolar geometry [9], given that the facial expression does not change significantly. The mathematical representation of the epipolar geometry, the fundamental matrix F, is given by

$$\mathbf{x}'^{\top} F \mathbf{x} = 0 \tag{1}$$

where $\mathbf{x} = (x, y, 1) \leftrightarrow \mathbf{x}' = (x', y', 1)$ is a corresponding point pair across two frames, given in homogeneous coordinates.

The fundamental matrix can be computed from 7 corresponding point pairs. Since there are outliers among the correspondences, we use the robust RANSAC algorithm [6]. By taking random samples of 7 pairs and evaluating the support for the corresponding fundamental matrices, the fundamental matrix with most support among the feature pairs is found. Point pairs which do not support this fundamental matrix are labeled as outliers. They can be removed from the point set since their displacements do not conform to the movement of the face. In Fig. 3, examples of such outlying features are given. When there are too few features left for fundamental matrix estimation (say < 25), they are removed and the face is not processed anymore.

(a) (b)

Figure 3: Two examples of outliers among the feature points. The cyan lines indicate the features' positions in the previous frame. The red dots are outliers and the green dots inliers w.r.t. the computed fundamental matrix. The outliers will be removed from the point set.

The number of iterations J needed to find an all-inlier sample can be determined from the required probability of success, i.e. the probability that at least one all-inlier sample is found in J iterations [6]. Let ε denote the outlier ratio in the data. If p is the required probability of success, e.g. 0.99, then we have the relation

$$p = 1 - (1 - (1 - \varepsilon)^7)^J \tag{2}$$

In case the value of ε is known, the number of iterations to be executed would be known beforehand. Usually, however, the outlier ratio is unknown and we must use an alternative way for determining the number of iterations. An efficient manner is to begin RANSAC with a conservative estimate of ε, for example 0.99, and adjust it during following iterations [9]. When a certain hypothesized model has the largest number of support points seen so far, we know that there are at least that many inliers in the data. This imposes a lower bound on $1 - \varepsilon$ for the data set, and we can adjust the remaining number of iterations J according to (2). The RANSAC algorithm using this strategy is given in Fig. 4. Here S_j is used to indicate the set of support points in iteration j, which are the points within a small distance T from the model. The largest support set found during the algorithm is denoted by S_{max}. The ratio $\frac{|S_{max}|}{n}$ with n the number of correspondences, is therefore a lower-bound for the inlier ratio $1 - \varepsilon$.

The re-estimation step at the end of the algorithm is performed to improve the accuracy of the final model. Since S_{max} will contain only points on a model, an ordinary least squares estimate will be sufficient.

- $j = 1, \quad J = \infty, \quad S_{max} = \emptyset$
- **while** $j < J$ **do**
  - Randomly select a sample of 7 points from the data and compute the corresponding fundamental matrix.
  - Determine the set of support points S_j for the fundamental matrix by verifying which points are within threshold distance T.
  - **if** $|S_j| > |S_{max}|$ **then**
    - $J = \log(1-p) \cdot \log^{-1}\left(1 - \left(\frac{|S_j|}{n}\right)^7\right)$
    - $S_{max} = S_j$
  - **end if**
  - $j = j + 1$
- **end while**
- Re-estimate the fundamental matrix based on the largest support set S_{max}.

Figure 4: The RANSAC algorithm for fundamental matrix estimation.

2.5 Warp - per frame

The previous SR frame will be warped into the coordinate frame of the current frame. This ensures that the SR frame always follows the current LR frame. For the warp only positions around feature positions are considered, i.e. detected faces containing feature points are being warped while the rest of the image is left unaffected. This is justified since the primary interest is the increased resolution of faces. The displacements for the pixel positions are obtained by simply averaging the displacements for the neighboring feature points. Let us denote the total set of feature correspondences between frame $n-1$ and n by $\mathbf{x}_{n-1} \leftrightarrow \mathbf{x}_n$ where $\mathbf{x} = (x,y)$. We then create a reversed dense flow map $f(\mathbf{x})$ and accompanying feature counter $n(\mathbf{x})$ by the accumulation process

$$\begin{aligned} &f(\mathbf{x}) = 0 \quad \forall\, \mathbf{x} \\ &\textbf{for each } \mathbf{x}_{n-1} \leftrightarrow \mathbf{x}_n \textbf{ do} \\ &\quad f([\mathbf{x}_n]) \leftarrow f([\mathbf{x}_n]) + (\mathbf{x}_{n-1} - \mathbf{x}_n) \\ &\quad n([\mathbf{x}_n]) \leftarrow n([\mathbf{x}_n]) + 1 \\ &\textbf{end for} \end{aligned} \tag{3}$$

where $[\mathbf{x}]$ denotes the rounded coordinate values. Subsequently, both arrays are convolved with kernel k

$$f(\mathbf{x}) \leftarrow k * f(\mathbf{x}) \qquad n(\mathbf{x}) \leftarrow k * n(\mathbf{x}) \tag{4}$$

where k is chosen as a square-shaped 21x21 kernel with unity elements, though other choices for k are also possible. The array $n(\mathbf{x})$ will eventually contain the number of feature points which fall within the support region of k centered at $\mathbf{x}$. The final flow map is created by

$$f(\mathbf{x}) \leftarrow \frac{f(\mathbf{x})}{n(\mathbf{x})} \quad \forall\, \mathbf{x} \tag{5}$$

After $f(\mathbf{x})$ for the LR frame is constructed in this way, it is interpolated with SR factor z and its values multiplied by z to yield the flow map in SR dimensions. The warped SR frame is then obtained by bicubic interpolation of SR frame $n-1$ at the positions indicated by $f(\mathbf{x})$. An additional step that is taken during the warp, is that at positions where $f(\mathbf{x}) = 0$ the SR pixels are directly copied from the bilinear interpolated LR frame. Since the warping from previous frames can introduce persistent artefacts in the images, a reset of the background is periodically required. It is unlikely that a face is affected in this way, since faces are assumed to be constantly moving.

2.6 SR algorithm - per frame

The SR method we use is from [10], which computes the difference between the observed LR frame I_{LR} and the simulated LR frame from the (warped) SR frame I_{SR}. The difference is then backprojected onto the SR frame. In particular, the SR frame is obtained by iterative application of

$$I_{SR} \leftarrow I_{SR} + \lambda g * ((I_{LR} - (g * I_{SR}) \downarrow z) \uparrow z) \tag{6}$$

where g, z and λ are the camera's point spread function (in SR dimensions), the SR factor and a weighting factor, respectively. The downward arrow represents subsampling of the

image, and the upward arrow upsampling with the insertion of zeros. The value of g has been chosen heuristically, and we have taken $\lambda = 0.2$. We update the SR frame according to (6) over 10 iterations. At the start of the video sequence, the initial SR frame is the bilinear interpolated LR frame 1. At frame n it is the warped SR frame $n-1$.

3 Experimental results

The SR scheme has first been applied to a video sequence from the NRC-IIT Facial Video Database [8], which contains compressed video sequences of size 160x120 pixels at 20 fps recorded by a webcam. The point spread function g was chosen as a Gaussian with size 7x7 and standard deviation 1.5. Only a single face and accompanying point set was allowed to be tracked in the sequence, since otherwise the point density would be much higher due to multiple detections. The threshold for including a point pair in the support set of a fundamental matrix was 0.5 pixel. In Fig. 5 the original, bilinearly interpolated and SR frames are shown with $z = 2$ for different frames in a sequence. An improvement of the SR images w.r.t. the original and interpolated images can be observed.

The effect of outlier rejection is shown in Fig. 6, where the SR result of a frame is also shown with omission of the steps in Section 2.4. Outlying feature points then remain in the point set and continue to produce misalignments, until they can not be tracked anymore or the total point set becomes too small.

In order to perform a quantitative measurement of the performance, we have downsampled the sequence from Fig. 5 to 80x60 pixels and subsequently run the algorithm with $z = 2$. In this way the mean squared error (MSE) of the results with the original frames can be computed. The MSE values are only calculated for the center image square of 20x20 pixels, which is most of the time enclosed by the face. The results for the first 150 frames are shown in Fig. 7. Indicated are the MSE values for the bilinearly interpolated frames w.r.t. the original frames, and for the proposed SR method w.r.t. the original frames. There are four periods when the face is covered with sufficient feature points to allow SR; here the MSE is smaller than for the interpolated frames. Due to face movement the feature points are lost over time, and only re-initialized when the face is detected again. The peaks where the MSE for the SR result exceeds that for the interpolated frames, are due to too large detection rectangles so that feature points are initialized outside of the face.

Although the effect of SR in the sequences of Fig. 5 is noticeable, it is not necessary for recognition of individuals; the original frame is large enough to perform recognition. A scene where recognition of individuals is considerably more difficult is shown in Fig. 8. This scene is taken from [1] which contains recordings made in a public area. The data is recorded at 25 fps with frame size 384 x 288 pixels, and is made available in JPEG format. In Fig. 8, the largest face is cropped from the frame, and shown separately together with the interpolated and SR faces. Since the face is 17x21 pixels in size, it is about the smallest face that can be found with the detector. We have used for g a Gaussian with standard deviation 1.2 in this example. It is true that reliable recognition from the SR image is still difficult, but the SR image does give more pronounced face features than the original or interpolated frames.

(a) original frames

(b) interpolated frames

(c) SR frames

Figure 5: The original, interpolated and super resolved faces for different frames of a webcam sequence.

4 Discussion

A SR scheme has been described for improving face images. The scheme relies on a face detector, so that optical flow vectors can be computed per face in the scene. It was proposed to use the epipolar geometry for rejecting outliers among the flow vectors. With the removal of such motion errors, the resulting SR image was improved.

In case persons move their head very fast or change their expression, the motion estimates may still be erroneous and the SR result deteriorates. Another issue is rotation of the head which may occlude some face parts or the whole frontal face area.

Currently, the SR scheme does not add feature points to already detected faces, so that the point set will eventually loose all its points while the face may still be visible. However, the addition of new points is not trivial, especially when the point set covers only a part of the face.

(a) No outlier rejection (b) Outlier rejection

Figure 6: A SR frame without (a) and with (b) outlier rejection.

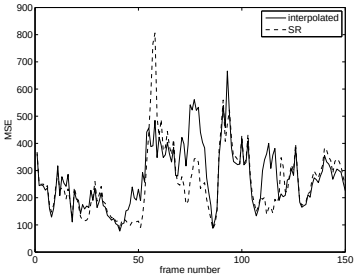


Figure 7: Mean squared errors for the center square of the interpolated and SR frames.

In addition to using the epipolar geometry merely for outlier rejection, it may also be used for inlier correction. The inlying points can be repositioned to fit the calculated geometry even better. Furthermore, a dense flow field may be obtained by exhaustive searching along the epipolar lines.

References

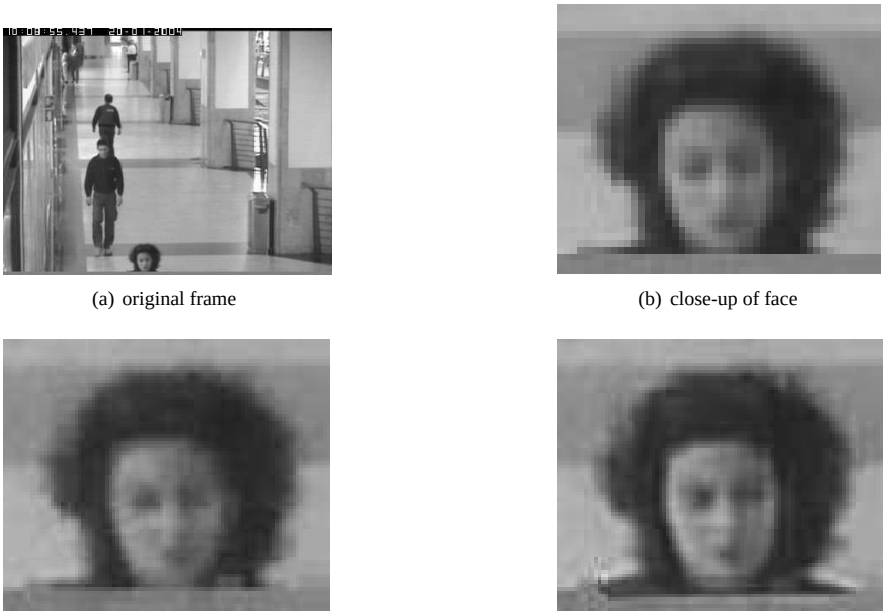
[1] EC funded caviar project / IST 2001 37540. http://homepages.inf.ed.ac.uk/rbf/CAVIAR/.

[2] Open computer vision library, Windows version beta 5 (21-7-2005). available from http://opencvlibrary.sourceforge.net/.

[3] S. Baker and T. Kanade. Hallucinating faces. Technical Report CMU-RI-TR-99-32, Robotics Institute, Carnegie Mellon University, 1999.

[4] S. Baker and T. Kanade. Super-resolution optical flow. Technical Report CMU-RI-TR-99-36, Robotics Institute, Carnegie Mellon University, 1999.

[5] J.-Y. Bouguet. Pyramidal implementation of the lucas kanade feature tracker. Included in OpenCV library.



(a) original frame (b) close-up of face

(c) interpolated image (d) SR image

Figure 8: SR performed on a face seen by a surveillance camera.

[6] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[7] R. Fransens, C. Strecha, and L. Van Gool. A probabilistic approach to optical flow based super-resolution. In *The workshop on Generative-Model Based Vision*, 2004.

[8] D.O. Gorodnichy. Video-based framework for face recognition in video. In *Proc. of the Second Canadian Conference on Computer and Robot Vision*, 2005.

[9] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2003.

[10] M. Irani and S. Peleg. Improving resolution by image registration. *CVGIP: Graphical Models Image Processing*, 53(3):231–239, 1991.

[11] K. Jia and S. Gong. Multi-modal tensor face for simultaneous super-resolution and recognition. In *IEEE International Conference on Computer Vision*, 2005.

[12] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. CVPR*, 2001.

Fibre Centred Tensor Faces

Bernard Tiddeman, Meng Yu and David Hunter
School of Computer Science,
University of St Andrews
bpt@cs.st-and.ac.uk, yumeng@cs.st-and.ac.uk, dwh@cs.st-and.ac.uk

Abstract

In this paper we present a reformulation of the tensorface analysis method and produce a model that is simpler (i.e. has fewer parameters), is more compact (i.e. has tighter distributions) and is less ambiguous (i.e. no 2 sets of parameters synthesise the same data vector). This is achieved by simply subtracting the fibre (row, column, etc) mean from each fibre of the training data before performing PCA analysis. Centring of tensor data via subtraction of the whole set mean is commonly used as a preprocessing step, but the fibre-centring algorithm presented here has not been suggested previously for tensorface analysis. We show how the new formulation allows an approximate linear analysis with a considerable speed improvement over previous methods. In addition, the centring allows simpler truncation of parameter vectors leading to an even more compact model. The new method is tested on image synthesis and analysis and in a simple face recognition task, in which it out performs non-centred multilinear analysis.

1 Introduction

Multilinear analysis has proved a useful tool in facial image analysis because of its ability to separate different sources of variation such as pose, expression, identity and lighting. While centring via subtraction of the whole set mean is frequently used as a preprocessing step, the choice of centring algorithm can lead to different results and more or less efficient algorithms. In this work we present a centring approach designed for face analysis. The centring approach used is to centre each "fibre" before unfolding the tensor, removing variation due to factors that are more easily accounted for using other axes of the face tensor. For example, using an identity-expression tensor, we would subtract each subject's average (across expressions) from the individual expression images and perform the expression part of the analysis on the resulting "identity free" tensor. This is repeated along the identity axis, removing the average of each expression from each individual expressive image, leaving "expression free" identity data. In this paper we first review the related literature, then explain the proposed method in detail and finally present preliminary results that demonstrate that this kind of processing not only leads to more appropriate parameters (for improved face recognition) it also leads to a faster analysis algorithm.

2 Literature Review

Multilinear analysis was introduced to the computer vision community by Vasilescou and Terzopoulos [1-4]. Multilinear data represent the natural extension from scalars (0-

D tensors), through vectors (1D tensor) and matrices (2D tensors) to general n-dimensional data. For face data the images are usually scan converted into a 1-D vector (vectorised), with the data vector representing 1 dimension of the face tensor and other attributes varying along the other axes e.g. along one axis identity, along another expressions and along another could be different lighting. To synthesise a new data vector from this block it seems intuitively reasonable to use one set of weights along each axis i.e. one for the identity axis, another along the expression axis and a third set of weights along the lighting axis. Hence the linear weightings of the example data vectors are given as the outer product tensor of the weights along each axis of the data block. This leads to a considerable reduction in the number of parameters required over a linear model. For example, using 10 identities, 10 expressions and 10 lightings would only require 30 parameters (10+10+10) instead of 1000 (10x10x10) using a linear combination of the training data vectors. The reconstruction from such a data set is therefore given by:

$$t = \sum_{i=0}^{I} \sum_{j=0}^{J} \cdots \sum_{k=0}^{K} a_i b_j \cdots c_k t_{ij\cdots k} \qquad (1)$$

where t is the reconstructed data vector, $a_i, b_j, \ldots, c_k$ are the model parameters and $t_{ij\ldots k}$ are the model's components.

As with linear face analysis, it has proved useful to perform additional (multi)linear processing on the input data. The standard decomposition is to factor the n-way tensor into N orthogonal matrices plus a core tensor. The usual method of construction is to unfold the tensor into a matrix along each axis, perform Singular Value Decomposition (SVD) on the unfolded matrix and then take the left orthogonal matrix. This is known as high-order SVD (HOSVD) or Tucker's method [5]. The HOSVD method does not yield unique output parameters and the core tensor is not diagonal. Alternatives have been proposed, for example Parallel Factor Analysis (PARAFAC), which is also known as Canonical Decomposition (CANDECOMP) [6][7], in which the data is modelled as the sum of N outer product tensors. This has the attractive properties of a diagonal core tensor and that the decomposition is unique, but the generating matrices are not orthogonal. The development of PARAFAC/CANDECOMP was motivated by a need to discover fundamental properties of the original data. For example in chemometrics it has been used to discover the pure spectra of mixed samples. Although one can think of applications in face analysis in which the primary aim is to discover fundamental properties of the original data (e.g. to find basic components of expressions) the aim in most of the existing face research is for compact, efficient and effective analysis and synthesis methods, and so the use of more efficient orthogonal decomposition remains attractive.

In order to analyse an example data vector several methods have been proposed. In their original tensorface work [1] Vasilescou and Terzopoulos proposed finding the optimal set of identity vectors for each set of non-identity parameters (expression, lighting, viewpoint). The optimal set of these weights that best matches the target are considered the best match. This needs to be repeated across the database to find the overall best match. This method has two drawbacks, the first is that it is close to an exhaustive search and therefore slow and the second is that it does not allow for interpolated values of the non-identity parameters e.g. lighting between two of the training lighting examples.

The ideal would be to find the parameters that minimise (1) in the least squares sense i.e.

$$\chi^2 = \sum_{x=0}^{X} \left(\sum_{i=0}^{I} \sum_{j=0}^{J} \cdots \sum_{k=0}^{K} a_i b_j \cdots c_k T_{ij\cdots k}(x) - T(x) \right)^2 \qquad (2)$$

This is a highly non-linear equation and difficult to solve explicitly. In [3] Vasilescou et al approximated the solution using a two step minimisation i.e. first they find the best linear parameters, $r_{ij\ldots k}$ that minimise:

$$\chi^2 = \sum_{x=0}^{X} \left(\sum_{i=0}^{I} \sum_{j=0}^{J} \cdots \sum_{k=0}^{K} r_{ij\cdots k} T_{ij\cdots k}(x) - T(x) \right)^2 \qquad (3)$$

and then they find the best rank 1 approximation to the Tensor $r_{ij\ldots k}$ using HOSVD. This is not guaranteed to find the best solution to (3), as other authors have noted [8], but seems a reasonable approximation.

Others have solved (2) using iterative methods. Park and Savvides [9] use the additional restriction that the magnitude of each of the weight vectors should be 1 (i.e. $\sum_i a_i^2 = \sum_j b_j^2 = \cdots = \sum_k c_k^2 = 1$). The resulting equations are solved iteratively using a Lagrange multiplier to include the additional constraint. This additional constraint is required because the original tensorface model is non unique, i.e. two (or more) sets of parameters can give the same output e.g. scaling the a_i by α and the b_j by $1/\alpha$ produces the same result. It is this non-uniqueness of the parametric description that we believe leads to under performance of the original tensor-face SVD model. Because the identity parameters are only known subject to some linear multiple, recognition is based on maximising the normalised scalar product between the known and test identity parameters. It is quite possible that two very different face images could have the same normalised identity parameters, depending on the values of the other parameters. Conversely, it is possible for the same face to have quite different parameters that lie along a line in the identity parameter space.

One difference between the tensorface analysis proposed by Vasilescou et al and the frequently used PCA analysis of facial images e.g. [12][13], is that in the latter the mean is explicitly subtracted from the set before performing the PCA analysis (e.g. using SVD or Jacobi rotations to find the Eigenvectors and Eigenvalues of the covariance matrix). This makes sense from the perspective of both the dimensionality of the linear space spanned by the data and from a Gaussian model of the distribution of the data. Given N non-collinear points in some (generally higher) M dimensional space, N-1 direction vectors are required to span the space, plus a single position vector within the linear space. Hence a space defined by N points only requires at most N-1 parameters to span the linear subspace. A convenient point to choose as the position vector within the space is the mean, as this minimises the size of the distribution.

Bro and Smilde [10] have considered the effects of centring and also scaling of the data in multilinear analysis, but they do not consider the situation in which the fibre means

are retained in the model along every dimension. Lu et al [11] have used a centred multilinear approach for gait analysis, but again the centring is only performed once along a single "samples" axis. It is not clear how their method could be extended to include appropriate centring along multiple sample axes, such as different styles of gait.

In this paper we adapt the HOSVD method with the addition of fibre centring. First we outline the mathematics of the reformulated representation and then describe methods for the analysis of an input vector and synthesis of an output vector. Finally we test the new method on a face recognition task and compare the performance with a non-centred multilinear method.

3 Method

3.1 Tensor construction

We propose a simple modification to the multilinear analysis of face images in which the mean is explicitly subtracted before processing. The mean we use in this context is the "fibre" mean (we will use fibre to refer to any 1D array extracted parallel to the Cartesian axes e.g. rows, columns, etc.) Removing the fibre mean leaves difference vectors that minimise the variation within the fibre. For example, using an identity-expression-lighting set we would remove the identity and expression variation from the lighting fibres, the identity and lighting variation from the expression fibres and the expression and lighting variation from the identity fibres. The hope is to minimise the variation due to known factors when processing along each axis.

For each axis of the tensor we first unfold the tensor along that axis and then multiply by the matrix [***I-M***], where each element of ***M*** is equal to *1/N*. This subtracts the fibre average from each sample. Next we perform PCA analysis on the resulting matrix to find the rotation matrix $\boldsymbol{U}_i$ and variances w_i. At most *N-1* of the variances will be non-zero, so we can discard at least the smallest component. We construct the matrix $\boldsymbol{U}_i'$ whose first row is an averaging vector and the remaining rows calculate the rotated difference components as follows:

$$U_i' = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & u_{00} & \cdots & u_{N0} \\ \vdots & & \ddots & \\ 0 & u_{0N-1} & \cdots & u_{NN-1} \end{bmatrix} \begin{bmatrix} 1/N & 1/N & \cdots & 1/N \\ 1-1/N & -1/N & \cdots & -1/N \\ \vdots & & \ddots & \vdots \\ -1/N & -1/N & \cdots & 1-1/N \end{bmatrix} \qquad (4)$$

where $\{u_{jk}\} = \boldsymbol{U}_i$. We perform the analysis on the original data along each axis, and then as a final step multiply the original tensor by the matrices $\boldsymbol{U}_i'$ along each direction. No processing is performed along the direction of the data vector. An alternative option is to perform the decomposition along all axes and then to undo the transform along the data vector using the inverse operator. The inverse operator $\boldsymbol{U}_i'^{-1}$ is given by:

$$U_i'^{-1} = \begin{bmatrix} 1 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 1 & \cdots & 0 \\ \vdots & & & \ddots & \vdots \\ 1 & 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & u_{00} & \cdots & u_{0N-1} \\ \vdots & & \ddots & \\ 0 & u_{N0} & \cdots & u_{N-1N} \end{bmatrix} \qquad (5)$$

3.2 Image synthesis

In order to reconstruct a single data vector (e.g. an image) we use an equation of the same form as used previously, i.e.:

$$\boldsymbol{t} = \sum_{i=0}^{I} \sum_{j=0}^{J} \cdots \sum_{k=0}^{K} a_i b_j \cdots c_k \boldsymbol{t}_{ij\cdots k} \qquad (6)$$

except that the first component of each parameter vector is 1, i.e. $a_0 = b_0 = ... = c_0 = 1$. The component $\boldsymbol{t}_{00...0}$ is the overall mean of the entire multilinear set and is given a constant weighting of 1. Hence the model has one less free parameter along each axis, and so is a simpler model. In addition because of the normalisation the distribution of parameter values are tighter and so truncation of the components can be justified. Truncation of the data has previously been performed using an iterative approach that tries to minimise the reconstruction error while preserving orthonormality of the decomposition vectors [4]. The centring scheme proposed here ensures that the truncated tensor model is optimal in the sense that the component with the smallest distribution from the row mean is discarded.

3.3 Image analysis

In this paper we propose 2 methods for estimating the parameters of our updated model given in equation (2). In the first we follow [3] and approximate the solution by first finding the linear estimates that solve equation (3). Because the weight of the overall mean is 1 we subtract the component $\boldsymbol{t}_{00..0}$ before minimising. The least squares solution of (3) for our model is given by:

$$\underline{\boldsymbol{r}} = (\boldsymbol{T}^t \boldsymbol{T})^{-1} \boldsymbol{T}^t (t - t_{00\cdots0}) \qquad (7)$$

where $\underline{\boldsymbol{r}}$ is the vector containing the vectorised version of the tensor $R = \{r_{ij...k}\}$ and $\boldsymbol{T}$ is the matrix whose columns are the data vectors of the tensor T excluding the mean (i.e. the tensor T unfolded along the data axis, less the mean).

In previous work HOSVD has been used to factor R into the best rank-1 approximation and find the outer product vectors. For the new model we note that the weights on the origin adjacent edges of the linear weight tensor should be equal to the parameter vectors we seek i.e.

$$r_{i0\cdots0} = a_i, r_{0j\cdots0} = b_j, \cdots, r_{00\cdots k} = c_k \qquad (8)$$

As an additional optimisation we do not need to calculate all of the linear components explicitly, as these can be "projected out". We only need the rows of the matrix $\boldsymbol{P} = (\boldsymbol{T}^t\boldsymbol{T})^{-1}\boldsymbol{T}^t$ that correspond to the origin adjacent edge components of $\underline{\boldsymbol{r}}$ in order to get a fast (linear) estimate of the model parameters. Once the matrix $\boldsymbol{P}$ has been calculated only $O(I+J+ ... +K)$ dot products are required to analyse an input image rather than $O(IJ...K)$ dot products required to compute the entire tensor R. Also this method does not require the additional work of using HOSVD to find the best rank 1 approximation of the tensor R and factor it into vectors.

If additional accuracy is found necessary we propose using an alternating least squares (ALS) approach. An initial estimate of the parameters can be found using equation (7) and then the parameters along each axis are re-estimated, keeping the others fixed i.e. estimate parameters a_i keeping b_j ... c_k fixed, then estimating parameters b_j, keeping a_i ... c_k fixed etc. This involves collapsing the model using the fixed components e.g. if we are optimising the parameters a_i the components are given by:

$$\boldsymbol{t}'_{\boldsymbol{i}} = \sum_{j=0}^{J} \cdots \sum_{k=0}^{K} b_j \cdots c_k \boldsymbol{t}_{\boldsymbol{ij}\cdots\boldsymbol{k}} \qquad (9)$$

and the model is given by:

$$\boldsymbol{t} = \boldsymbol{t}'_0 + \sum_{i=1}^{I} a_i \boldsymbol{t}'_i \qquad (10)$$

which is solved for a_i. This process can be repeated until convergence (i.e. the parameters remain stable).

3.4 Face recognition method

In this section we describe how to find the best match from a set of example faces to a previously unseen facial image. In order to compute the best match, previous authors have used the maximum normalised scalar product between the known identity components and those estimated for the test image, i.e.

$$\hat{\boldsymbol{a}} \cdot \hat{\boldsymbol{b}} = \frac{\sum a_i b_i}{\sqrt{\sum a_i^2 \sum b_i^2}} \qquad (11)$$

is maximised, where $\boldsymbol{a}$ and $\boldsymbol{b}$ are the identity parameter vectors for two subjects. The normalised scalar product was used because of the unknown scaling of the parameters. In contrast the values of the fibre-centred tensor parameters are known precisely, so we have explored alternative recognition methods. One option is to build a Gaussian model of the identity parameters for each subject and calculate the maximum probability over all subjects i.e. maximise:

$$p(\boldsymbol{a}) = \frac{1}{(2\pi)^{n/2}|\boldsymbol{C}_b|} \mathrm{e}^{-\frac{1}{2}(\boldsymbol{a}-\bar{\boldsymbol{b}})' \boldsymbol{C}_b^{-1} (\boldsymbol{a}-\bar{\boldsymbol{b}})} \qquad (12)$$

where $\boldsymbol{a}$ is the identity parameter vector of the test image and $\boldsymbol{b}$ and $\boldsymbol{C}_b$ are the mean and covariance matrix that define the Gaussian distribution of identity parameters for the training face that we are testing against. The problem with this method is that the distribution of identity parameters for an individual in the training data is usually zero or very close to zero. This could be seen as a case of over fitting of the model to the training data, and the same is true of the non-centred HOSVD, save for the unknown sign of the normalised components. We could populate the Gaussian model by using additional example images of the same subjects used to train the tensor model but as this

requires an additional image set, we instead use the simpler method of finding the closest mean to the test subject i.e. we seek to minimise:

$$\sum_{i=0}^{I} (a_i - \bar{b}_i)^2 \qquad (13)$$

where $\boldsymbol{a}$ is the test identity vector and $\boldsymbol{b}$ is the mean identity vector for the training image we are comparing against.

4 Results

In this section we present the results of our reformulated tensorface methods and compare them with the original algorithms. First we describe our data preprocessing, then we demonstrate the results of analysis and synthesis of face images. Finally we use the method for a face recognition task and compare the results with standard HOSVD recognition.

The test set of images need to contain standardised images that vary along a number of dimensions (such as identity, pose, lighting). Many of the existing test sets, such as the FERET set are not suitable for multilinear training, as the poses etc are not standardised across subjects. Other more suitable images sets exists, such as the Yale database and the Weizman image set. We use two example sets for training and testing, one set of our own images and one from a standard collection. The first is a bi-linear set in which 7 subjects are performing 16 different viseme (visual phonemes). The peak viseme images were extracted from video by hand. The images in this set are face-on images with consistent lighting. We use an out-of-set (neutral) image for testing. The second set is a tri-linear subset of the Weizmann face database. The images in the subset are face-on images and show 15 identities, 3 lightings and 3 expressions. A fourth image of each subject with a neutral expression and different lighting is used as the test set.

4.1 Data pre-processing

In this work we decompose the facial image data into a shape component and a shape normalised image component. Each image was delineated automatically using an active appearance model (AAM) [11][12] and any errors were corrected by hand. The average of the feature points across the set was calculated and was shifted and scaled to fit in a window of a user specified size (for the experiments shown here we used a 100 by 150 pixel image). The delineated feature template of each subject was then normalised to this average using best fit scaling, rotation and translation before being vectorised and placed in the input shape tensor. The input images were warped into the normalised average shape using piecewise linear warping over a triangulation of the template feature points. All but the face region was masked and then the resulting images were vectorised and added to the input image tensor.

4.2 Model Building Example

Figure 1 shows the decomposition process applied to the subset of the Weizmann face set. Only 3 of the 15 identities used are shown. The input data tensor comprising the shape normalised example images is on the left. The result of processing along each axis in turn with the matrix $\boldsymbol{U}_i'$ is shown going from left to right. The original data and

averaged components are unmodified for display and the processed difference components are scaled and shifted so that mid-grey equals zero for display.

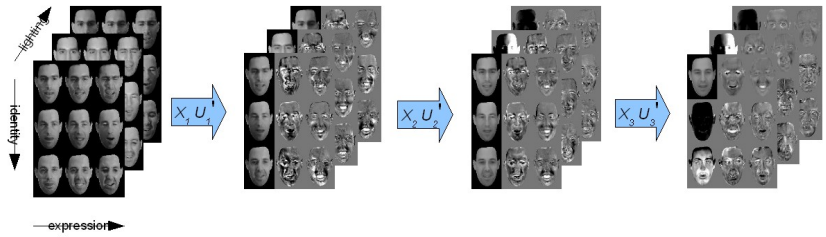


Figure 1: Example of the decomposition process on a set of (shape normalised) face images.

4.3 Analysis and Synthesis Examples

Figure 2 shows the synthesis process. The synthesis process forms a weighted sum of the centred multilinear data (left). The weights (centre) are formed from the outer product tensor of the parameter vectors. Each parameter vector starts with the value 1 and so the contribution of the overall mean is always 1. Figure 3 illustrates the linear analysis process. Figure 4 shows some examples of face synthesis from parameters found using the linear approximation. Examples are given with and without truncation of the model.

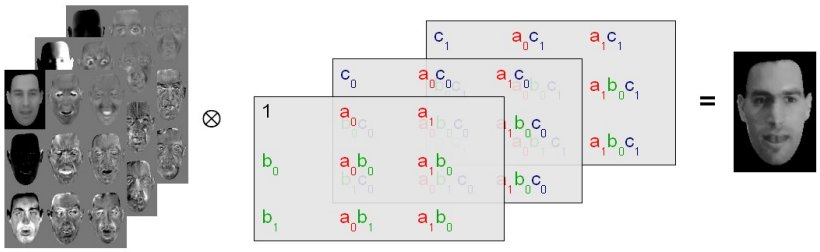


Figure 2: A diagram of the face synthesis method for the normalised image components.

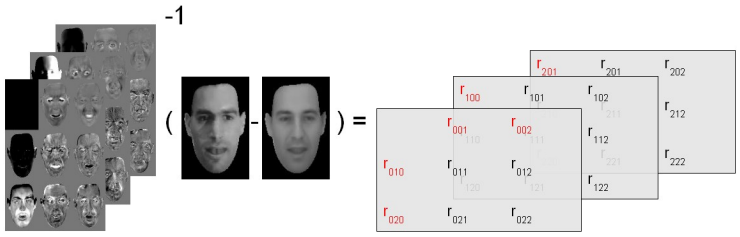


Figure 3: Diagram showing the (approximate) linear analysis process. Only the components of the pseudo inverse matrix (left) corresponding to the linear coefficients shown in red (right) need to be used.

Figure 4: Face synthesis by analysis examples from the two data sets used. In each set: left original, centre 100% and right 95% of variance in each parameter explained in the two data sets.

4.4 Face Recognition Experiments

We tested the centred tensor face model's performance on a simple face recognition task using our two face sets for training. We tested the model's ability to correctly assign an identity to previously unseen images of the same individuals as used in the training set. For the viseme set this was a neutral expression image and for the Weizmann set it was a neutral image with different lighting. For the centred model we used the approximate linear analysis and truncated the model parameters to explain 95% of the variance in the training data along each axis. For the viseme set this reduced the viseme parameters from 15 to 12 in shape and from 15 to 13 in colour but did not truncate the identity parameters. For the Weizmann subset the identity parameters were reduced from 15 to 12 in both shape and colour and the lighting and expression parameters were not truncated. The results are shown in Table 1.

	Non-centred	Centred
Viseme	71%	100%
Weizmann	67%	100%

Table 1. Results of the simple face recognition experiments.

5 Conclusions and Future Work

In this paper we have developed a reformulation of the multilinear PCA analysis of face images and have demonstrated improved recognition performance over previous HOSVD face recognition in a simple test. The improved performance is almost certainly due to the unknown scaling of the identity parameters in the original HOSVD

method requiring the use of the normalised scalar product or normalised nearest neighbour for recognition. This normalisation can project different subjects into a similar part of the parameter space. In contrast in the centred method the parameters are unique and so can be used directly.

Future work will focus on testing the method on larger datasets and for other recognition tasks e.g. expression recognition. We would also like to compare our new method with multilinear independent component analysis (MICA) and explore an ICA version of the algorithm presented here.

References

[1] M.A.O. Vasilescu and D. Terzopoulos. *Multilinear analysis for facial image recognition*. In Proc. Int. Conf. on Pattern Recognition, volume 2, pages 511–514, Quebec City, August 2002.

[2] M.A.O. Vasilescu and D. Terzopoulos. *Multilinear analysis of image ensembles: Tensorfaces*. In Proc. European Conf. on Computer Vision (ECCV 2002), pages 447–460, Copenhagen, Denmark, May 2002.

[3] M.A.O. Vasilescu and D. Terzopoulos, *Multilinear Independent Components Analysis*, Proc. Computer Vision and Pattern Recognition Conf. (CVPR '05),

[4] M.A.O. Vasilescu, D. Terzopoulos, *Multilinear Subspace Analysis for Image Ensembles*,' Proc. Computer Vision and Pattern Recognition Conf. (CVPR '03), Vol.2, Madison, WI, June, 2003, 93-99.

[5] P.M. Kroonenburg, *Three-mode principal component analysis. Theory and applications*. DSWO Press, Leiden, 1983.

[6] R.A. Harshman, *Foundations of the PARAFAC procedure: Model and conditions for an 'explanatory' multi-mode factor analysis.* UCLA Working Papers in phonetics, 16 (1970) 1.

[7] J.D. Carroll and J. Chang, *Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition*. Psychometrika, 35 (1970) 283.

[8] I. Macedo, E. V. Brazil, and L. Velho. *Expression Transfer between Photographs through Multilinear AAMs*. In Proceedings of SIBGRAPI 2006 - XIX Brazilian Symposium on Computer Graphics and Image Processing, Manaus, October 2006

[9] S.W. Park, M. Savvides, *Estimating Mixing Factors Simultaneously in Multilinear Tensor Decomposition for Robust Face Recognition and Synthesis*, cvprw, p. 49, 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06), 2006.

[10] Rasmus Bro and Age K. Smilde, *Centering and scaling in component analysis*, Journal of Chemometrics, 2003, 17, 16-33.

[11] Haiping Lu , K.N. Plataniotis and A.N. Venetsanopoulos, *Multilinear Principal Component Analysis of Tensor Objects for Recognition*, 18th International Conference on Pattern Recognition (ICPR'06), pp. 776-779.

[12] T. F. Cootes, G. J. Edwards, and C. J. Taylor. *Active appearance models*. In 5th European Conference on Computer Vision (ECCV 1998), pages 484–498, June 1998.

[13] I. Matthews and S. Baker. *Active appearance models revisited*. International Journal of Computer Vision, 60(2):135–164, Nov. 2004.

Binary Co-occurrences of Weak Descriptors

Martin Winter and Horst Bischof
Graz University of Technology, Austria *

Abstract

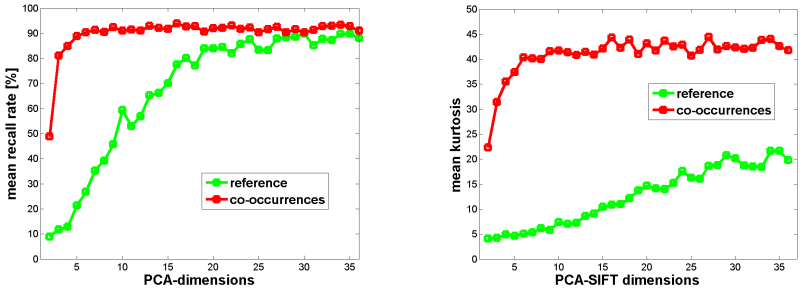
This paper demonstrates that a reliable and efficient object recognition system based only on binary joint occurrences of quantized descriptors can be built. Specifically, we show that a high recognition performance can be obtained even with very weak (non discriminative) descriptors. The binary joint occurrence representation despite being high dimensional is very sparse and therefore efficient. In order to obtain reliable joint occurrences we present a fast hierarchical quantization algorithm. We illustrate our results using different descriptors (PCA-SIFT, Spin images, SIFT) on a challenging, specific object recognition task and consider the scaling behavior of the method.

1 Introduction

In the last decade, object recognition tasks based on local features gained more and more interest by the computer vision community. A lot of different approaches have been proposed and recent evaluations have shown satisfactory performance on specific as well as generic object recognition challenges (*e.g*. [2, 5, 8, 16]). To increase the discriminative power of the approaches, the algorithms tend to be more and more complex thus requiring increasing computational power, runtime and necessary amount of memory. For example, key-point detectors are made robust against image distortions, viewpoint- and illumination changes [14, 15], descriptors are driven toward enhanced discriminative power, distinctiveness and invariance [12, 14] and complex selection and decision algorithms such as support vector machines or boosting algorithms [6, 23] have been applied to several recognition systems. It is evident that approaches considering single key-points and descriptors (bag of key-point approaches such as *e.g*. [5]) have severe limitations by not taking spatial neighborhood relations into account. Therefore, recent work has investigated spatial relations among key-points. Generative models use the spatial relations directly in modeling the assembly of object parts. One prominent example is the 'constellation model' of Fergus *et al*. [9], a fully connected probabilistic model of a few object parts trained to identify object classes on unsegmented, cluttered scenes. Simpler 'relational models' have been proposed by *e.g*. Crandall and Huttenlocher (k-fan model) [4] or Fergus *et al*. (star-shape model) [10]. Carneiro and Lowe [3] proposed an approach which does not rely on strict spatial models, but allows to adapt the spatial relations specifically to the underlying object properties. Another important approach, which primarily tries

*This work has been funded by the Austrian Joint Research Project Cognitive Vision under sub-projects S9103-N04 and S9104-N04 and it has been partially supported by the European Union Network of Excellence, MUSCLE, under contract FP6-507752.

to overcome the limitations on the number of distinct features in the model, is the one from Bouchard and Triggs [2], where the authors proposed a probabilistic, hierarchical approach on features and spatial relations of object parts. Another class of approaches uses spatial relations as additional information. Those spatial relations are either used to improve the discriminative power of features directly (*e.g.* 'Hyperfeatures' [1] by Agarwal and Triggs) or they are used for verification of tentative matches (*e.g.* 'semi-local constraints' by Schmid and Mohr [21]). Another example for spatial relations in a second processing step is the texture recognition system proposed by Lazebnik *et al.* [13] where spatial neighborhood statistic is used for final texture classification. There are also other interesting approaches that use spatial relations such as the one from Sivic *et al.* [7] (classification with probabilistic Latent Semantic Analysis (pLSA) in a 'bag of words' model) or the joint spatial relation of boundary fragments from Opelt *et al.* [19] . In summary all these papers (and many others) demonstrate, that spatial relations can significantly improve recognition results.



(a) (b)

Figure 1: Co-occurrence and reference algorithm mean recall rates (a). Mean kurtosis values of our object recognition task for decreasing PCA-SIFT descriptor size (b).

This evidence triggered the work in this paper where we take an extreme case. The only information we consider is the joint occurrence or co-occurrence of quantized descriptors. Moreover, we only represent the presence or absence of these co-occurrences which leads to a binary representation. The intuition behind is, that the co-occurrence of descriptors is a very discriminative feature because it is very unlikely, that two descriptors co-occur just by chance. In particular, we show that due to the distinctiveness of co-occurrences we can work even with very weak descriptors which are not discriminative by themselves. The following experiment demonstrates this fact (full details are explained in the experimental section 3). We use a straight forward, state of the art recognition setup: key-point extraction, descriptor calculation, quantization and matching. As descriptor we selected PCA-SIFT [12] proposed by Ke and Sukthankar, because we can control the weakness of that descriptor by consecutively reducing the descriptor dimension. The main result of the experiment is depicted in Figure 1a and shows the mean recall rate as function of the diminishing information (PCA-dimension). First one sees as expected, that using a single descriptor decreasing the PCA dimension decreases the recognition rate. More important, using co-occurrences of descriptors the recognition rate remains high despite the increasing weakness of the descriptor. This indicates, that the co-occurrence approach is not only much more discriminative, but can deal with 'weaker' descriptors still preserving

a high recognition rate. In the following section we present the details of the binary co-occurrence approach and the proposed recognition system. In the experimental section 3 we describe a set of experiments which foster our assumptions, show that co-occurrences also increase the reliability and significance of an object recognition task and consider the scaling behavior.

2 Binary Co-occurrence overview

Our system has four main components, namely: 1. identification of key-points and calculation of a descriptor, 2. clustering the descriptors of the training database features to obtain a visual vocabulary, 3. nearest neighbor assignment of features and building the binary co-occurrence matrix representation and 4. matching the co-occurrence matrices against the database representation. An overview of the components and their interaction is depicted in Figure 2.

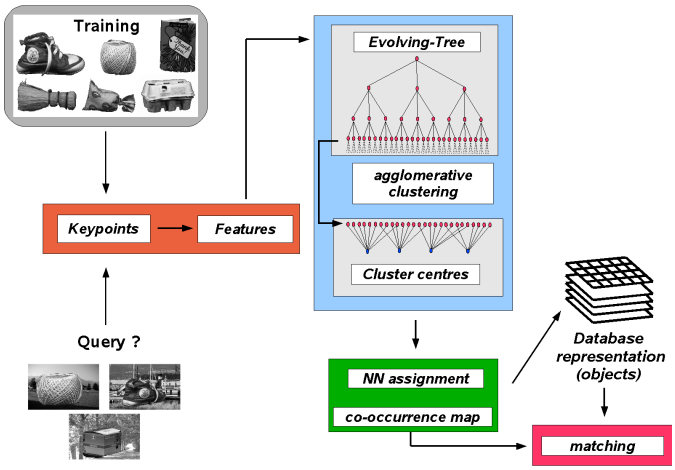


Figure 2: Main components of our co-occurrence recognition system and combination of the Evolving-Tree with the agglomerative clustering algorithm.

2.1 Obtaining the visual vocabulary

To cope with the large number of descriptors obtained by state of the art key-point detectors, we need to cluster them efficiently. Standard clustering methods such as k-means or agglomerative clustering cannot handle such a large amount of descriptors in an efficient manner. Therefore we use a novel combination of two clustering algorithms with different properties. The first one is a tree based variant of a self organizing map, namely the Evolving-Tree proposed by Pakkanen *et al.* [20]. The Evolving-Tree randomly takes training samples from a data-set and uses nearest neighbor assignments until a leaf node is reached. Similar to self organizing maps, the leaf node location is updated with a Kohonen

learning rule and once a certain number of leaf node members is reached, the node splits up into a predefined number of child nodes. Due to its simplicity the Evolving-Tree is very fast (we need only about 50 seconds to cluster 200K samples on a Intel Xeon 2.80GHz CPU), but unfortunately the clustering is very imprecise. In accordance with Pakkanen *et al*. [20] we have observed, that the global nearest neighbor property is not fulfilled for most points indexed by the tree. Although the problem can be reduced by pruning the tree, the results remained unsatisfactory. Thus we use a second clustering algorithm and re-cluster the prototypes of the obtained cluster centers in the pruned Evolving-Tree. This second clustering algorithm is a modification of 'agglomerative hierarchical clustering' similar to that one used in [16]. The main advantage of this algorithm is the fact, that we have to select only the tolerated dissimilarity of two points belonging to the same cluster in the feature space. So we can combine the advantages of the two algorithms: (1) the speed of the Evolving-Tree allows us to roughly quantize a feature space even with a very large number of samples in feasible time and (2) the agglomerative clustering algorithm guarantees the necessary similarity of samples assigned to a single cluster. So starting with typically about 200K descriptors we use for creating the visual vocabulary, we end up in a typically size of about $k = 10K$ clusters in a few minutes. The tree also speeds up the search during the recognition as we quickly traverse the Evolving-Tree down to the leaf node and make use of the obtained mapping from agglomerative clustering in the training step.

2.2 Training and recognition of objects, the co-occurrence matrix

Once we have obtained the cluster-centers, training of the database is straight forward. Every object is presented to the recognition system and key-points and descriptors are calculated. Every descriptor is assigned to the best matching cluster center, so that for every interest-point a cluster label is stored. To calculate the co-occurrence matrix, we identify the n nearest neighbors (typically $n = 3$) in image space for every key-point. Thus, every co-occurrence is identified by a pair of cluster centers, which we insert into a two dimensional co-occurrence matrix. The rows and columns of the co-occurrence matrix are the cluster labels of the two key-point descriptors. The co-occurrence matrix is very sparsely populated. Typically only $1 - 2‰$ of the possible co-occurrences are assigned. Thus it is possible, to store only the binary information, whether a co-occurrence is present or not. This binary coding and a sparse storage schema allows us to reduce the necessary amount of memory to a minimum. To build the full representation for a single object (multiple viewpoints), all the co-occurrences of the training images are entered in one single matrix. Therefore, we have exactly one co-occurrence matrix per object trained and the final dimensionality of the training data representation is given by the squared number of cluster centers times the number of objects represented in the database. For the recognition of objects with the co-occurrence matrices we follow the same steps as for training, but for a single query image and obtain a single co-occurrence matrix for each one. The matching procedure itself is deliberately kept very simple. We calculate the matching score for every object representation of the training database by simply AND operation of the co-occurrence matrices and count the number of resulting matches. So in fact, the matching is only a simple maximum voting of congruent co-occurrences in the binary matrices.

3 Experiments

The main purpose of our experiments is to show, that with co-occurrences we can obtain respectable and reliable recognition results even with a very weak and fast descriptor. We do that by calculating 'recall'-rates for a challenging recognition scenario up to 900 different objects on substantially cluttered background. We first present the database and the reference system we compare our results with, and then several experiments highlighting different aspects of our approach.

3.1 The object recognition task

For our first experiment, we use a subset of the first 50 objects from the publicly available Amsterdam Library of Object Images (ALOI) [11]. Recognition approaches based on local features require a reasonable number of DoG key-points detected on the objects surface. Therefore we sorted the objects of the whole ALOI with respect to their sum of key-points detected, and shifted the upper and lower 50 ones to the end of the database. Thus objects with no or too few key-points as well as those showing an exceptionally high number are not taken into account. To capture enough variances in the appearances, during the training stage we present 12 views (every 30^o) per object to the system. This is similar to the approach of Murase and Nayer [18], but in contrast to them we use a larger amount of objects, can deal with occlusions and work on substantial background clutter - important necessities for a realistic recognition scenario. So in the recognition step, the system has to recognize an object presented in different viewpoint angles and projected to challenging background images (see Figure 3 for some examples). For all of our experiments we use the publicly available binaries of Lowe's 'Difference of Gaussian' (DoG) detector [14] to obtain rather accurate key-points with high repeatability.

Figure 3: Some examples of objects and partially occluded objects from ALOI database.

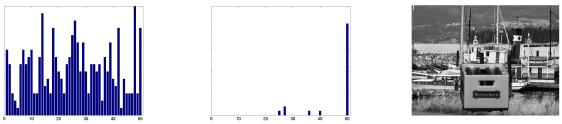
3.2 The reference recognition system

To show the impact of the introduced co-occurrence approach, we compare a simple standard voting scheme against our approach. We try to keep the algorithmic parts as much as possible identical for the two systems (Figure 2). The only differences between the reference and co-occurrence recognition system is the representation of the objects (cluster centers versus binary co-occurrence matrix) and the matching method. For the reference system we use an inverse histogram to object voting where every key-point descriptor in a query image is assigned to its nearest neighbor cluster and every cluster votes for one or more objects (similar to [22]). The object with the maximum number of votes is selected. It is essential for the reference recognition system to use only very distinctive feature to cluster assignments, rejecting features near the separation plane of different cluster centers. We follow the approach of Lowe in [14] and use the ratio between the first and

second nearest neighbor cluster as a good measure for that property (a distance ratio of $r = 0.8$ gives the best results in our experiments).

3.3 Decreasing information content of the descriptor

The goal of this experiment is to show, that even by decreasing the information content of a descriptor, our approach still keeps feasible recall rates and the matching reliability is higher than for the reference recognition system. As a descriptor we selected PCA-SIFT, a variant of a gradient based descriptor introduced by Ke and Sukthankar [12]. The descriptor is a PCA decomposition of 2 orthogonal gradient images and it gives similar performance rates like SIFT, even if the dimensions are reduced down to 20 [17]. In order to simulate continuously diminishing information content, we can successively remove the less important values of the PCA-projection indicated by the magnitude of the original eigenvalues. We trained it for each dimensionality reduction factor separately as we want to allow the special 'adaptation' of the Evolving-Tree to the desired dimensionality. The main results of the experiment (see Figure 1a) have already been mentioned in the introduction. For decreasing PCA dimension (weaker descriptor) the recognition rates decrease as expected, but the single descriptor approach breaks down much earlier. This is in accordance to the observations of Ke and Sukthankar proposing a dimensionality of $n = 20$ as a good trade-off between matching speed, storage requirements and good recognition results [12]. The co-occurrence approach shows excellent stability and recall performance. It is able to produce nearly perfect recognition results for only 5 or 6 PCA dimensions. Even for 2 PCA dimensions the recall rate is rather high. This experiment indicates, that the co-occurrence approach is not only much more discriminative, but also can deal with 'weaker' descriptors while keeping high recall rates. Furthermore we investigate the significance of the voting processes for the reference and co-occurrence algorithm. Even by manual inspection of the voting histograms, the higher significance of the co-occurrence approach is obvious (Figure 4). In order to get a quantitative estimate of



(a) (b) (c)

Figure 4: Example of voting histogram for the reference algorithm (a) and our co-occurrence approach (b). The corresponding object is shown in column (c).

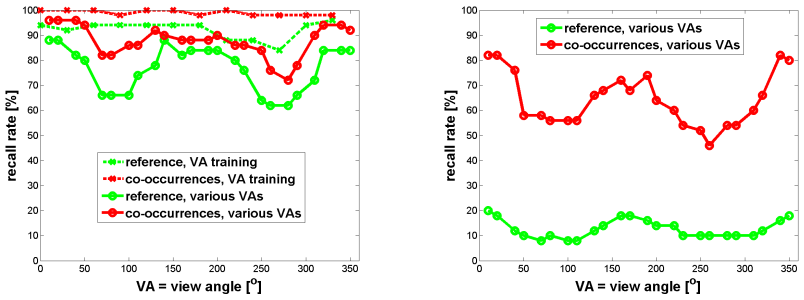
the 'voting significance' we calculate the kurtosis of the voting histogram function. The kurtosis is a statistical measure for the 'peakedness' of a function. The 'ideal kurtosis' (impulse function) for a perfect voting histogram of 50 different objects is $k_{opt} = 48.0204$. In Figure 1b one can see the kurtosis of the reference and our co-occurrence algorithm for a decreasing number of PCA components. The difference of the 'voting significance' measure is obvious even for the highest number of PCA dimensions tested ($n = 36$). So taking into account the results of Figure 1a we can state, that even for comparable recognition rates, the significance of the voting histograms is much higher for the co-occurrence approach resulting in increased reliability of recognition.

3.4 Co-occurrences with SIFT descriptors

In this experiment we show the impact of co-occurrences for standard SIFT-keys as they are a standard descriptor for recognition systems. The recall rates for different viewing angles (VA) can be seen in Figure 5a. We split the recognition results into different curves for the viewing angles originally presented to the system during training (dotted line) and the slightly rotated views of the object on the background image (full line). The recall rate for the co-occurrence approach on 'trained' objects (objects presented to the system in the training stage) is nearly 100%. The improvement obtained by co-occurrences is only about 10% to 15% for various viewpoint angles. Although the improvement introduced through co-occurrences is obvious, the difference is not really large, but SIFT descriptors are already very distinctive features, so that for such a 'strong' descriptor the high performance rates even for a simple voting algorithm are not surprising.

3.5 Spin-Images as an example for a 'weak descriptor'

In this experiment we want to demonstrate the power of our approach applied to a 'real' weak descriptor. A disadvantage of many gradient based descriptors is the necessity to normalize the patches to the principal direction of the region to obtain rotational invariance. Besides the fact, that orientation estimation is not always correct, sometimes more than one orientation is predominant, so that even two or more variants of a patch have to be considered [14]. So for a weak descriptor we prefer a simple, rotational invariant descriptor ideally working on gray values so that we can completely avoid gradient calculation and orientation estimation. A good candidate for such a weak, rotational invariant descriptor is a 2D modification of the original spin-image descriptor proposed by Lazebnik *et al*. [13]. It can be calculated very efficiently and has shown low(!) performance compared to most other descriptors as *e.g*. shown in [17]. Nevertheless in combination



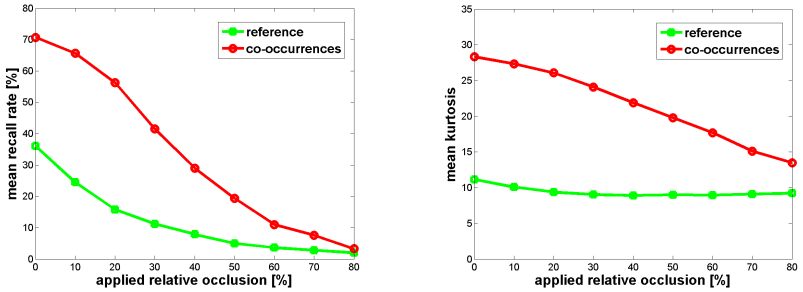
(a) (b)

Figure 5: Recall rates for our co-occurrence algorithm and the reference system on first 50 objects of ALOI database. (a) SIFT descriptor and (b) 'spin images' as a weak descriptor.

with co-occurrences even that poor descriptor can lead to encouraging recognition rates. Figure 5b shows the results of our experiment. As expected, the recall rate for our method consistently outperforms the reference algorithm and an average performance boost of about 60% is obtained. The decrease of performance between 50 to 100 and 250 to 300 degrees is due the fact, that most of the object are captured from the largest side.

3.6 Recognition of partially occluded objects

To support the claim on the improvement of recognition reliability, we repeated the experiments of the last section with the spin image descriptor, but with varying partial occlusions of query images. The occlusions are simulated by transparent rectangles, covering a substantial part of the objects appearance (Figure 3). Transparency means, that we can view the random background on the occluded parts of the objects. In order to avoid biasing the results for elongated objects in certain directions, we have averaged the recall rates for vertical and horizontal occlusions. Figure 6 shows the mean recall rates (a) and kurtosis



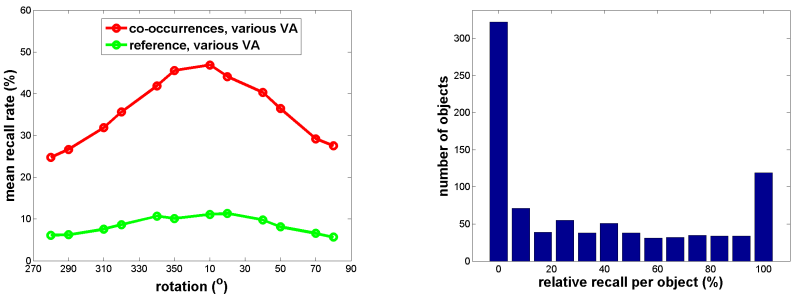
(a) (b)

Figure 6: Mean recall rates (a) and kurtosis values (b) for a different amount of occlusions.

values of the voting histograms (b) for a different amount of occlusions on the 'sea' background. Besides the already demonstrated fact, that our co-occurrence approach shows consistently better recall rates, one can see the different curvature of the recall function in the range from zero up to 40% recall rate. Thus, the relative decay of recognition rates with respect to the initial values is significantly higher for the standard algorithm those indicating the higher stability of our proposed approach. The same is evident by means of the constantly higher kurtosis values for co-occurrences.

3.7 Scaling behavior and vocabulary generalization

In order to investigate the scaling behavior of our approach, we extended the recognition task to the first 900 objects (6 views every 30^o trained) of the ALOI database. Furthermore we use only the first 100 objects to learn the visual vocabulary which results consistently in about 10K clusters as in the experiments before. Thus, there is no special adaptation of the vocabulary to the whole database (generalization of vocabulary). Figure 7a shows the obtained recall rates for the SIFT descriptors. The difference between the reference system and our proposed algorithm is obvious. The experiment consistently supports the improvement of recall rates and higher matching reliability of the feature co-occurrence algorithm even for much more objects. Nevertheless, despite considering the natural increase of possible mismatches due to the database size, the recall rate is not that high and needs some further consideration. Figure 7b depicts a histogram illustrating the number of objects for a specific relative recall rate. 15% of the objects are perfectly recognized, while about one third of the objects in the database do not work at all. Inspecting those objects we have found the following problems: (1) there are many objects which do not



(a) (b)

Figure 7: (a) Recall rates for our co-occurrence algorithm and the reference system (SIFT). (b) Number of objects with a specific relative recall rate of the first 900 objects.

have a sufficient number of key-points that are detected (*e.g.* no texture, very small objects). (2) some objects can be discriminated only using color, since we use a gray-level based representation these objects are indistinguishable for our method. (3) highlights from the acquisition set-up introduce consistent false matches.

4 Summary and conclusion

In this paper we have demonstrated, that only binary co-occurrences of quantized descriptors are sufficient to build a reliable and efficient object recognition system. Besides the simplicity of the approach the main novelty is the fact, that we do not use the spatial relations for verification of tentative matches or enrichment of other extensive object representations. In our approach we use the co-occurrences alone for object representation and recognition. In order to obtain reliable co-occurrences we present a fast hierarchical quantization algorithm and by limiting the representation of an object to a solely binary representation we can heavily reduce the storage requirements and limit the matching to a very simple and crude algorithm. In a central experiment we could verify, that even by substantially decreasing the information content of a PCA-SIFT descriptor our approach still keeps feasible recall rates and the matching reliability for the feature co-occurrence is much higher, than for a standard recognition approach. Furthermore we have shown, that the approach works well with 'spin-images' as an recognition example for a weak but easy computable descriptor, even for substantial occlusion. A final scaling experiment demonstrates the performance increase caused by the novel representation with respect to the bag of key-points approach. The conclusion of the paper is the fact, that instead of further increasing the distinctiveness of certain descriptors and applying computationally expensive matching algorithms, one can also take simple 'weak' descriptors and achieve the distinctiveness by the simple concept of binary feature co-occurrence.

References

[1] Ankur Agarwal and Bill Triggs. Hyperfeatures – multilevel local coding for visual recognition. In *Proceedings ECCV*, volume 3951 of *LNCS*, pages 30–43. Springer, 2006.

[2] Guillaume Bouchard and Bill Triggs. Hierarchical part-based visual object categorization. In *Proceedings CVPR*, volume 1, pages 710–715, 2005.

[3] Gustavo Carneiro and David Lowe. Sparse flexible models of local features. In *Proceedings ECCV*, volume 3953 of *LNCS*, pages 29–43, 2006.

[4] David J. Crandall and Daniel P. Huttenlocher. Weakly supervised learning of part-based spatial models for visual object recognition. In *Proceedings ECCV*, pages 16–29. Springer, 2006.

[5] Gabriella Csurka, Christopher R. Dance, Lixin Fan, JuttaWillamowski, and Cedric Bray. Visual categorization with bags of key-points. In *Proceedings SLCV*, 2004.

[6] Jamie Shotton et al. *TextonBoost*: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *Proceedings ECCV*, pages 1–15. Springer, 2006.

[7] Josef Sivic et al. Discovering objects and their location in images. In *Proceedings ICCV*, volume 1, pages 370–377, 2005.

[8] Mark Everingham et al. The 2005 pascal visual object classes challenge. In *Selected Proceedings of the First PASCAL Challenges Workshop (LNAI), Springer-Verlag*, April 2006.

[9] Rob Fergus, Pietro Perona, and Andrew Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proceedings CVPR*, volume 2, pages 264–271, June 2003.

[10] Rob Fergus, Pietro Perona, and Andrew Zisserman. A sparse object category model for efficient learning and exhaustive recognition. In *Proceedings CVPR*, 2005.

[11] Jan Mark Geusebroek, Gertjan Burghouts, and Arnold Smeulders. The Amsterdam library of object images. *International Journal of Computer Vision*, 61(1):103–112, January 2005.

[12] Yan Ke and Rahul Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *Proceedings CVPR*, volume 2, pages 506–513, 2004.

[13] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Affine-invariant local descriptors and neighborhood statistics fortexture recognition. In *Proceedings ICCV*, pages 649–655, 2003.

[14] David Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[15] Jiri Matas, Ondrei Chum, U. Martin, and Tomas Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings BMVC*, volume 1, pages 384–393, 2002.

[16] Krystian Mikolajczyk, Bastian Leibe, and Bernt Schiele. Local features for object class recognition. In *Proceedings ICCV*, pages 1792–1799. IEEE Computer Society, October 2005.

[17] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[18] Hiroshi Murase and Shree K. Nayar. Visual learning and recognition of 3d objects from appearance. *International Journal of Computer Vision*, 14(1):5–24, Januar 1995.

[19] Andreas Opelt, Axel Pinz, and Andrew Zisserman. Incremental learning of object detectors using a visual alphabet. In *Proceedings CVPR*, volume 1, pages 3–10, 2006.

[20] Jussi Pakkanen, Jukka Iivarinen, and Erkki Oja. The evolving tree-analysis and applications. *IEEE Transactions on Neural Networks*, 17(3):591–603, 2006.

[21] Cordelia Schmid and Roger Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:530–535, 1997.

[22] Josef Sivic and Andrew Zisserman. Video Google: A text retrieval approach to object matching invideos. In *Proceedings ICCV*, October 2003.

[23] Hongming Zhang, Wen Gao, Xilin Chen, and Debin Zhao. Object detection using spatial histogram features. *Image and Vision Computing*, 24(4):327–341, April 2006.

Capturing Correlations Among Facial Parts for Facial Expression Analysis

Caifeng Shan, Shaogang Gong, and Peter W. McOwan
Department of Computer Science
Queen Mary, University of London
Mile End Road, London E1 4NS, UK
{cfshan, sgg, pmco}@dcs.qmul.ac.uk

Abstract

Capturing and analyzing the correlations among facial parts are important for interpreting facial behaviors precisely. In this paper, we exploit Canonical Correlation Analysis (CCA) to model the correlations of facial parts for facial expression analysis. We propose a Matrix-based Canonical Correlation Analysis (MCCA) for better correlation analysis on 2D image or matrix data in general. Extensive experiments have shown that compared to the traditional CCA, MCCA models more accurately correlations among image data with more compact representation using much fewer canonical factors.

1 Introduction

Automatic facial expression analysis has attracted much attention in recent years [13]. As facial muscles are contracted in unison to display expressions, different facial parts have strong correlations. Capturing and analyzing the correlations among facial parts are important for interpreting facial expressions precisely. Most of the existing work on facial expression analysis [4, 14, 2, 11] did not explicitly model the correlations between facial parts. In this paper, we employ Canonical Correlation Analysis (CCA) (Section 2), a statistical technique that is well suited for relating two sets of signals, to model correlations of facial parts for facial expression analysis.

CCA was developed [8] for measuring linear relationships between two vector variables. It finds pairs of base vectors (i.e., canonical factors) for two variables such that the correlations between the projections of the variables onto these canonical factors are mutually maximized. Recently CCA has been applied to computer vision problems [1, 12, 7, 10, 5]. Borga [1] adopted CCA to find corresponding points in stereo images. Melzer *et al.*[12] applied CCA to model the relation between an object's poses with raw brightness images for pose estimation. Like Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), CCA also reduces the dimensionality of the original variables, since only a few factor pairs are normally needed to represent the relevant information. However, they serve different purposes: whilst PCA aims to minimize the reconstruction error and LDA derives a discriminant function that maximizes between-class scatter and minimize within-class scatter, CCA seeks directions for two sets of variables to maximize their correlations, so it is better suited for regression tasks. It has been shown that CCA outperforms PCA for regression tasks [12]. Recently Donner *et al.*[5] presented a fast Active Appearance Model search algorithm, which uses reduce-rank regression estimates obtained by CCA, instead of standard linear least-square regression estimates.

In the existing work, when applying CCA to image data, the original two-dimensional images have to be reshaped into one-dimensional vectors, as the traditional CCA is based on the vector-space model. However, this matrix-to-vector operation leads to two main problems. Firstly, the intrinsic 2D structure of image matrices is removed, so the spatial information stored therein is discarded. CCA based on these vectors can not fully capture correlations among the original 2D image data. Secondly, each image sample is modeled as a high-dimensional vector so that a large number of training samples are needed to yield a reliable estimation of the underlying data distribution. However, in reality, very limited number of training data are usually available. Actually these problems are shared by other subspace methods such as PCA and LDA. Recently some methods have been proposed to extend these vector-based methods for 2D matrices or high-order tensors [16, 17, 3, 15]. However, all these existing matrix-based methods were developed for learning in one set of variables, and not suited for measuring relationships between two set of variables.

To address these problems, we introduce a novel Matrix-based Canonical Correlation Analysis (MCCA) for better correlation analysis of 2D image or matrix data in general (Section 3). MCCA takes a 2D matrix based data representation rather than the 1D vector based representation in classical CCA. So the collection of data is represented as a set of matrices, instead of a single large matrix. MCCA seeks canonical factors in two dimensions to maximize the correlations between two sets of matrices. Unlike classical CCA, there is no closed-form solution for the optimization problem in MCCA. Instead, we propose an iterative solution with a convergence proof. We evaluate the proposed MCCA in capturing correlations of facial parts for facial expression analysis (Section 4). Experimental results demonstrate that MCCA can better measure correlations in 2D image data, providing superior performance in regression and recognition tasks, whilst requiring much fewer canonical factors. We notice that more recently Zou *et al.*[18] introduced a 2DCCA by simply replacing the image vector with image matrix in computing the variance matrices. Their approach is different to ours both in concept and algorithmic design; moreover, they addressed the correlations between image sets and their label matrices, instead of two sets of images.

2 Canonical Correlation Analysis

Given two zero-mean random variables $\mathbf{x} \in R^m$ and $\mathbf{y} \in R^n$, CCA finds pairs of directions $\mathbf{w}_x$ and $\mathbf{w}_y$ that maximize the correlation between the projections $x = \mathbf{w}_x^T \mathbf{x}$ and $y = \mathbf{w}_y^T \mathbf{y}$ (x and y are called *canonical variates*). More formally, CCA maximizes the function:

$$\rho = \frac{E[xy]}{\sqrt{E[x^2]E[y^2]}} = \frac{E[\mathbf{w}_x^T \mathbf{x}\mathbf{y}^T \mathbf{w}_y]}{\sqrt{E[\mathbf{w}_x^T \mathbf{x}\mathbf{x}^T \mathbf{w}_x]E[\mathbf{w}_y^T \mathbf{y}\mathbf{y}^T \mathbf{w}_y]}} = \frac{\mathbf{w}_x^T \mathbf{C}_{xy} \mathbf{w}_y}{\sqrt{\mathbf{w}_x^T \mathbf{C}_{xx} \mathbf{w}_x \mathbf{w}_y^T \mathbf{C}_{yy} \mathbf{w}_y}} \quad (1)$$

where $\mathbf{C}_{xx} \in R^{m \times m}$ and $\mathbf{C}_{yy} \in R^{n \times n}$ are the *within-set covariance matrices* of $\mathbf{x}$ and $\mathbf{y}$, respectively, while $\mathbf{C}_{xy} \in R^{m \times n}$ denotes their *between-sets covariance matrix*. A number of at most $k = \min(m,n)$ canonical factor pairs $\langle \mathbf{w}_x^i, \mathbf{w}_y^i \rangle, i = 1, \ldots, k$ can be obtained by successively solving $\arg\max_{\mathbf{w}_x^i, \mathbf{w}_y^i} \{\rho\}$ subject to $\rho(\mathbf{w}_x^j, \mathbf{w}_x^i) = \rho(\mathbf{w}_y^j, \mathbf{w}_y^i) = 0$ for $j = 1, \ldots, i-1$, i.e., the next pair of $\langle \mathbf{w}_x, \mathbf{w}_y \rangle$ are orthogonal to the previous ones.

The maximization problem can be solved by setting the derivatives of Eqn. (1), with

respect to $\mathbf{w}_x$ and $\mathbf{w}_y$, equal to zero, resulting in the eigenvalue equations as:

$$\begin{cases} \mathbf{C}_{xx}^{-1}\mathbf{C}_{xy}\mathbf{C}_{yy}^{-1}\mathbf{C}_{yx}\mathbf{w}_x = \rho^2\mathbf{w}_x \\ \mathbf{C}_{yy}^{-1}\mathbf{C}_{yx}\mathbf{C}_{xx}^{-1}\mathbf{C}_{xy}\mathbf{w}_y = \rho^2\mathbf{w}_y \end{cases} \tag{2}$$

Matrix inversions need to be performed in Eqn. (2), leading to numerical instability if $\mathbf{C}_{xx}$ and $\mathbf{C}_{yy}$ are rank deficient. Alternatively, $\mathbf{w}_x$ and $\mathbf{w}_y$ can be obtained by computing principal angles, as CCA is the statistical interpretation of principal angles between two linear subspace [6] (see [10] for details).

3 Matrix-based Canonical Correlation Analysis

We present an approach to perform canonical correlation analysis on 2-dimensional images or matrices in general. Given two matrix variables $\mathbf{A} \in R^{m\times n}$ and $\mathbf{B} \in R^{j\times k}$ (we assume the variables are both zero-mean), MCCA finds pairs of directions $\mathbf{v}_a \in R^m$, $\mathbf{w}_a \in R^n$, $\mathbf{v}_b \in R^j$ and $\mathbf{w}_b \in R^k$ that maximize the correlation between the projections $a = \mathbf{v}_a^T\mathbf{A}\mathbf{w}_a$ and $b = \mathbf{v}_b^T\mathbf{B}\mathbf{w}_b$. Mathematically, we can formulate this as the following maximization problem: find optimal $\mathbf{v}_a$, $\mathbf{w}_a$, $\mathbf{v}_b$ and $\mathbf{w}_b$ that maximize

$$\rho = \frac{E[ab]}{\sqrt{E[a^2]E[b^2]}} = \frac{E[\mathbf{v}_a^T\mathbf{A}\mathbf{w}_a\mathbf{w}_b^T\mathbf{B}^T\mathbf{v}_b]}{\sqrt{E[\mathbf{v}_a^T\mathbf{A}\mathbf{w}_a\mathbf{w}_a^T\mathbf{A}^T\mathbf{v}_a]E[\mathbf{v}_b^T\mathbf{B}\mathbf{w}_b\mathbf{w}_b^T\mathbf{B}^T\mathbf{v}_b]}} \tag{3}$$

Here $\mathbf{v}_a$ ($\mathbf{v}_b$) and $\mathbf{w}_a$ ($\mathbf{w}_b$) are canonical factors in two dimensions, acting as a two-sided linear transformation on the data in matrix form. To our knowledge, there is no closed-form solution for the maximization problem in Eqn. (3). A key observation, which leads to an iterative algorithm for the computation of $\mathbf{v}_a$, $\mathbf{w}_a$, $\mathbf{v}_b$ and $\mathbf{w}_b$, is stated in the following Lemma:

Lemma 1 *Let $\mathbf{v}_a$, $\mathbf{w}_a$, $\mathbf{v}_b$ and $\mathbf{w}_b$ be the optimal solution to the maximization problem in Eqn. (3), then*
(1) Given $\mathbf{w}_a$ and $\mathbf{w}_b$, $\mathbf{v}_a$ and $\mathbf{v}_b$ can be obtained as canonical factors of two variables $\mathbf{a}' \in R^m$ and $\mathbf{b}' \in R^j$, where $\mathbf{a}' = \mathbf{A}\mathbf{w}_a$ and $\mathbf{b}' = \mathbf{B}\mathbf{w}_b$.
(2) Given $\mathbf{v}_a$ and $\mathbf{v}_b$, $\mathbf{w}_a$ and $\mathbf{w}_b$ can be obtained as canonical factors of two variables $\mathbf{a}'' \in R^n$ and $\mathbf{b}'' \in R^k$, where $\mathbf{a}'' = \mathbf{A}^T\mathbf{v}_a$ and $\mathbf{b}'' = \mathbf{B}^T\mathbf{v}_b$.

Proof (1) $\mathbf{v}_a$, $\mathbf{w}_a$, $\mathbf{v}_b$ and $\mathbf{w}_b$ maximize Eqn. (3), which can be rewritten as

$$\rho = \frac{E[\mathbf{v}_a^T\mathbf{a}'\mathbf{b}'^T\mathbf{v}_b]}{\sqrt{E[\mathbf{v}_a^T\mathbf{a}'\mathbf{a}'^T\mathbf{v}_a]E[\mathbf{v}_b^T\mathbf{b}'\mathbf{b}'^T\mathbf{v}_b]}} \tag{4}$$

where $\mathbf{a}' = \mathbf{A}\mathbf{w}_a$ and $\mathbf{b}' = \mathbf{B}\mathbf{w}_b$. Hence, given $\mathbf{w}_a$ and $\mathbf{w}_b$, the maximum of Eqn. (4) is achieved by solving canonical correlation analysis on the variables $\mathbf{a}'$ and $\mathbf{b}'$ (by the definition of CCA in Eqn. (1)). So $\mathbf{v}_a$ and $\mathbf{v}_b$ can be obtained as canonical factors of $\mathbf{a}'$ and $\mathbf{b}'$.

(2) Similarly, Eqn. (3) can also be rewritten as

$$\rho = \frac{E[\mathbf{w}_a^T\mathbf{a}''\mathbf{b}''^T\mathbf{w}_b]}{\sqrt{E[\mathbf{w}_a^T\mathbf{a}''\mathbf{a}''^T\mathbf{w}_a]E[\mathbf{w}_b^T\mathbf{b}''\mathbf{b}''^T\mathbf{w}_b]}} \tag{5}$$

where $\mathbf{a}'' = \mathbf{A}^T\mathbf{v}_a$ and $\mathbf{b}'' = \mathbf{B}^T\mathbf{v}_b$. Hence, given $\mathbf{v}_a$ and $\mathbf{v}_b$, the maximum of Eqn. (5) is achieved by solving canonical correlation analysis on the variables $\mathbf{a}''$ and $\mathbf{b}''$. So $\mathbf{w}_a$ and $\mathbf{w}_b$ can be obtained as canonical factors of $\mathbf{a}''$ and $\mathbf{b}''$. This completes the proof of the lemma.

By the above Lemma, we present an iterative procedure for computing $\mathbf{v}_a$, $\mathbf{w}_a$, $\mathbf{v}_b$ and $\mathbf{w}_b$ as follows: given the initial choice of $\mathbf{w}_a$ and $\mathbf{w}_b$, we can compute $\mathbf{v}_a$ and $\mathbf{v}_b$ by computing canonical factors of $\mathbf{a}'$ and $\mathbf{b}'$; with the computed $\mathbf{v}_a$ and $\mathbf{v}_b$ (corresponding to the largest canonical correlation), we can then compute $\mathbf{w}_a$ and $\mathbf{w}_b$ by computing canonical factors of $\mathbf{a}''$ and $\mathbf{b}''$, and $\mathbf{w}_a$ and $\mathbf{w}_b$ (corresponding to the largest canonical correlation) will be used in next iteration. The procedure can be repeated until convergence. In this way, a number of at most $q = \min(m, j)$ left-side canonical factor pairs $\langle \mathbf{v}_a^1, \mathbf{v}_b^1 \rangle, \ldots, \langle \mathbf{v}_a^q, \mathbf{v}_b^q \rangle$ and a number of at most $p = \min(n, k)$ right-side canonical factor pairs $\langle \mathbf{w}_a^1, \mathbf{w}_b^1 \rangle, \ldots, \langle \mathbf{w}_a^p, \mathbf{w}_b^p \rangle$ can be obtained. The pseudo-code of the above iterative procedure is given in Algorithm 1, where `CCA(`$\mathbf{a}, \mathbf{b}$`)` computes the canonical factors and canonical correlations of the variables $\mathbf{a}$ and $\mathbf{b}$.

Algorithm 1: MCCA

1 Obtain initial choice $\mathbf{w}_a^{(0)}$ and $\mathbf{w}_b^{(0)}$ for $\mathbf{w}_a$ and $\mathbf{w}_b$, and set $\rho^{(0)} \leftarrow -1$ and $i \leftarrow 0$;
2 **repeat**
3 $\quad i \leftarrow i + 1$;
4 $\quad (\mathbf{v}_a^s, \mathbf{v}_b^s, \rho^s) \leftarrow$ `CCA(`$\mathbf{A} * \mathbf{w}_a^{(i-1)}, \mathbf{B} * \mathbf{w}_b^{(i-1)}$`)` ;
$\quad$ /* $s = 1, \ldots, q$ */
5 $\quad \mathbf{v}_a^{(i)} \leftarrow \mathbf{v}_a^1$, $\mathbf{v}_b^{(i)} \leftarrow \mathbf{v}_b^1$, $\rho^{(i)} \leftarrow \rho^1$;
6 $\quad (\mathbf{w}_a^t, \mathbf{w}_b^t, \rho^t) \leftarrow$ `CCA(`$\mathbf{A}^T * \mathbf{v}_a^{(i)}, \mathbf{B}^T * \mathbf{v}_b^{(i)}$`)` ;
$\quad$ /* $t = 1, \ldots, p$ */
7 $\quad \mathbf{w}_a^{(i)} \leftarrow \mathbf{w}_a^1$, $\mathbf{w}_b^{(i)} \leftarrow \mathbf{w}_b^1$, $\rho^{(i)} \leftarrow \rho^1$;
8 **until** $\rho^{(i)} - \rho^{(i-1)} < \varepsilon$;
9 $\mathbf{V}_a \leftarrow [\mathbf{v}_a^1, \ldots, \mathbf{v}_a^q]$, $\mathbf{V}_b \leftarrow [\mathbf{v}_b^1, \ldots, \mathbf{v}_b^q]$;
10 $\mathbf{W}_a \leftarrow [\mathbf{w}_a^1, \ldots, \mathbf{w}_a^p]$, $\mathbf{W}_b \leftarrow [\mathbf{w}_b^1, \ldots, \mathbf{w}_b^p]$;

3.1 Proof of Convergence

The convergence of MCCA follows, since correlation coefficient ρ is bounded between -1 and 1 from its definition, as stated in the following theorem:

Theorem 2 *The MCCA algorithm monotonically non-decreases the value of correlation coefficient* ρ*, hence it converges in the limit.*

Proof Given $\mathbf{w}_a^{(i-1)}$, $\mathbf{w}_b^{(i-1)}$ and $\rho^{(i-1)}$ obtained in Line 7, `CCA(`$\mathbf{A} * \mathbf{w}_a^{(i-1)}, \mathbf{B} * \mathbf{w}_b^{(i-1)}$`)` in Line 4 finds optimal $\mathbf{v}_a$ and $\mathbf{v}_b$ that maximize

$$\rho = \frac{E[ab]}{\sqrt{E[a^2]E[b^2]}} = \frac{E[\mathbf{v}_a^T \mathbf{A} \mathbf{w}_a^{(i-1)} {\mathbf{w}_b^{(i-1)}}^T \mathbf{B}^T \mathbf{v}_b]}{\sqrt{E[\mathbf{v}_a^T \mathbf{A} \mathbf{w}_a^{(i-1)} {\mathbf{w}_a^{(i-1)}}^T \mathbf{A}^T \mathbf{v}_a] E[\mathbf{v}_b^T \mathbf{B} \mathbf{w}_b^{(i-1)} {\mathbf{w}_b^{(i-1)}}^T \mathbf{B}^T \mathbf{v}_b]}} \tag{6}$$

Apparently, the value of $\rho^{(i-1)}$ is derived as

$$\begin{aligned}\rho^{(i-1)} &= \frac{E[ab]}{\sqrt{E[a^2]E[b^2]}} \\ &= \frac{E[\mathbf{v}_a^{(i-1)^T}\mathbf{A}\mathbf{w}_a^{(i-1)}\mathbf{w}_b^{(i-1)^T}\mathbf{B}^T\mathbf{v}_b^{(i-1)}]}{\sqrt{E[\mathbf{v}_a^{(i-1)^T}\mathbf{A}\mathbf{w}_a^{(i-1)}\mathbf{w}_a^{(i-1)^T}\mathbf{A}^T\mathbf{v}_a^{(i-1)}]E[\mathbf{v}_b^{(i-1)^T}\mathbf{B}\mathbf{w}_b^{(i-1)}\mathbf{w}_b^{(i-1)^T}\mathbf{B}^T\mathbf{v}_b^{(i-1)}]}}\end{aligned} \quad (7)$$

which is less or equal to the maximized canonical correlation that $\texttt{CCA}(\mathbf{A}*\mathbf{w}_a^{(i-1)}, \mathbf{B}*\mathbf{w}_b^{(i-1)})$ finds. so the derived $\rho^{(i)}$ in Line 5 is no less than $\rho^{(i-1)}$. With regard to the first iteration, given any initial choice $\mathbf{w}_a^{(0)}$ and $\mathbf{w}_b^{(0)}$, the canonical correlation $\rho^{(1)}$ derived by $\texttt{CCA}(\mathbf{A}*\mathbf{w}_a^{(0)}, \mathbf{B}*\mathbf{w}_b^{(0)})$ is no less than -1 ($\rho^{(0)}$). Therefore, the update of ρ in Line 5 do not decrease its value, since the computed ρ is locally optimal.

Similarly, given $\mathbf{v}_a^{(i)}$, $\mathbf{v}_b^{(i)}$ and $\rho^{(i)}$ obtained in Line 5, $\texttt{CCA}(\mathbf{A}^T*\mathbf{v}_a^{(i)}, \mathbf{B}^T*\mathbf{v}_b^{(i)})$ in Line 6 finds optimal $\mathbf{w}_a$ and $\mathbf{w}_b$ that maximize

$$\rho = \frac{E[ab]}{\sqrt{E[a^2]E[b^2]}} = \frac{E[\mathbf{v}_a^{(i)^T}\mathbf{A}\mathbf{w}_a\mathbf{w}_b^T\mathbf{B}^T\mathbf{v}_b^{(i)}]}{\sqrt{E[\mathbf{v}_a^{(i)^T}\mathbf{A}\mathbf{w}_a\mathbf{w}_a^T\mathbf{A}^T\mathbf{v}_a^{(i)}]E[\mathbf{v}_b^{(i)^T}\mathbf{B}\mathbf{w}_b\mathbf{w}_b^T\mathbf{B}^T\mathbf{v}_b^{(i)}]}} \quad (8)$$

Apparently, the value of $\rho^{(i)}$ in Line 5 is derived as

$$\rho^{(i)} = \frac{E[ab]}{\sqrt{E[a^2]E[b^2]}} = \frac{E[\mathbf{v}_a^{(i)^T}\mathbf{A}\mathbf{w}_a^{(i-1)}\mathbf{w}_b^{(i-1)^T}\mathbf{B}^T\mathbf{v}_b^{(i)}]}{\sqrt{E[\mathbf{v}_a^{(i)^T}\mathbf{A}\mathbf{w}_a^{(i-1)}\mathbf{w}_a^{(i-1)^T}\mathbf{A}^T\mathbf{v}_a^{(i)}]E[\mathbf{v}_b^{(i)^T}\mathbf{B}\mathbf{w}_b^{(i-1)}\mathbf{w}_b^{(i-1)^T}\mathbf{B}^T\mathbf{v}_b^{(i)}]}} \quad (9)$$

which is less or equal to the maximized canonical correlation that $\texttt{CCA}(\mathbf{A}^T*\mathbf{v}_a^{(i)}, \mathbf{B}^T*\mathbf{v}_b^{(i)})$ finds. So the update of ρ in Line 7 do not decrease its value too. Therefore, the MCCA optimization process monotonically non-decreases the ρ value, and converges in the limit. This completes the proof of the theorem.

The convergence of the MCCA algorithm was also confirmed experimentally. We show some examples of iterative learning in Fig. 1 and Fig. 2, where each example is for the learning on a different training set. We can observe that the value of ρ becomes stable after at most 20-30 iterations. We also found that any variation on the initial choice of $\mathbf{w}_a^{(0)}$ and $\mathbf{w}_b^{(0)}$ has almost no effect on convergence (as observed in Fig. 2). The fast and stable convergence keeps the training cost low.

3.2 Effect of the Initial Choice $\mathbf{w}_a^{(0)}$ and $\mathbf{w}_b^{(0)}$.

Theoretically, our solution to MCCA is only locally optimal. This solution depends on the initial choice $\mathbf{w}_a^{(0)}$ and $\mathbf{w}_b^{(0)}$. However, in practice this does not have any ill-effect. We conducted extensive experiments using different choices for $\mathbf{w}_a^{(0)}$ and $\mathbf{w}_b^{(0)}$, and found that, for image datasets, MCCA always converges to a similar (if not identical) solution regardless of the initial choice $\mathbf{w}_a^{(0)}$ and $\mathbf{w}_b^{(0)}$.

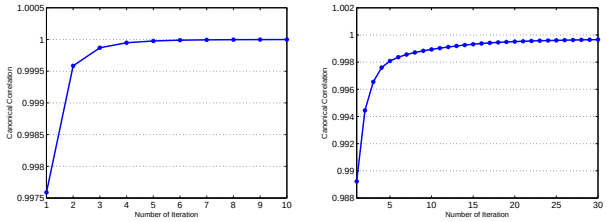


Figure 1: Convergence property of MCCA.

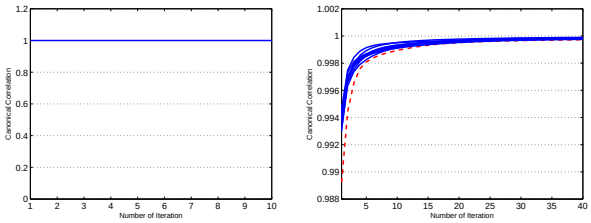


Figure 2: Sensitivity of MCCA to the initial choice $\mathbf{w}_a^0$ and $\mathbf{w}_b^0$: the ten solid curves correspond to the ten runs with random initializations, and the dash curve corresponds to $\mathbf{w}_a^0 = \mathbf{w}_b^0 = (1,0,\ldots,0)^T$ (the dash curve in the *left* side is identical with solid curves, so is not visible).

We show two typical results in Fig. 2, where the horizontal axis is the number of iterations and the vertical axis is the value of ρ. Each sub-figure is the results on a different training set. We run MCCA with 10 randomly generated $\mathbf{w}_a^{(0)}$'s and $\mathbf{w}_b^{(0)}$'s, and another initialization $\mathbf{w}_a^{(0)} = \mathbf{w}_b^{(0)} = (1,0,\ldots,0)^T$. For the left side of Fig. 2, we can observe that MCCA converges within two iterations for all eleven initial choices with the specified threshold ($\varepsilon = 10^{-5}$), and also converges to the same solution. In the right side of Fig. 2, MCCA converges slower. For all different initial choices, MCCA converges within 20-30 iterations with the threshold $\varepsilon = 10^{-5}$, and converges to very similar solutions. The difference between the values of final ρ is very small ($< 1.6 \times 10^{-4}$). These experiments demonstrate that, for image datasets, MCCA always converges to a similar (if not identical) solution regardless of the initial choice $\mathbf{w}_a^{(0)}$ and $\mathbf{w}_b^{(0)}$. We used the initial choice $\mathbf{w}_a^{(0)} = \mathbf{w}_b^{(0)} = (1,0,\ldots,0)^T$ in our experiments.

4 Experiments

As a case study, we investigate correlations between the mouth part (Mouth) and the right eye part (Eye) (as shown in Fig. 3). These two parts have strong and a range of correlations corresponding to facial expressions. We conducted experiments on the Cohn-Kanade database [9] and face expression image sequences we captured. We manually normalized the faces based on three feature points, centers of the two eyes and the mouth, using affine transformation. In the normalized facial images (110×150 pixels), the mouth part is 53×68 pixels, and the eye part is 45×51 pixels.

Figure 3: A case study on correlations between the mouth and the right eye facial parts.

4.1 Facial Parts Synthesis

We wish to reconstruct (synthesize) Mouth from Eye or vice versa using MCCA based regression. Specifically, to reconstruct image **B** from image **A**, we first employ MCCA to establish their relationship, finding optimal projection directions in the sense of correlation, and then map A to the leading canonical variates by discarding directions with low canonical correlation. Finally we perform regression of **B** by taking these leading canonical variates of **A**. The procedure of synthesis is as follows.

1. Compute the leading factor pairs $\mathbf{V}_a, \mathbf{W}_a, \mathbf{V}_b, \mathbf{W}_b$ from N pairs of samples $\mathscr{A} = \{\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_N\}$ and $\mathscr{B} = \{\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_N\}$.

2. Map $\mathbf{A}_i$ $(i = 1, \ldots, N)$ to the reduced correlation space $\widetilde{\mathbf{A}}_i = \mathbf{V}_a^T \mathbf{A}_i \mathbf{W}_a$.

3. Reshape 2D matrices $\widetilde{\mathbf{A}}_i$ and $\mathbf{B}_i$ to 1D vectors $\tilde{\mathbf{a}}_i$ and $\mathbf{b}_i$, and form data matrices $\widetilde{\mathbf{A}} = [\tilde{\mathbf{a}}_1, \ldots, \tilde{\mathbf{a}}_N]$ and $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_N]$; then compute the regression matrix $\mathbf{R} = (\widetilde{\mathbf{A}}^T)^{-1}\mathbf{B}^T$.

4. Given a new input $\mathbf{A}_{new}$, the corresponding $\mathbf{B}_{new}$ is reconstructed by:

$$\widetilde{\mathbf{A}}_{new} = \mathbf{V}_a^T \mathbf{A}_{new} \mathbf{W}_a, \quad \widetilde{\mathbf{A}}_{new} \rightarrow \tilde{\mathbf{a}}_{new} \tag{10}$$

$$\mathbf{b}_{new} = \mathbf{R}^T \tilde{\mathbf{a}}_{new}, \quad \mathbf{b}_{new} \rightarrow \mathbf{B}_{new} \tag{11}$$

Here $\widetilde{\mathbf{A}}_{new} \rightarrow \tilde{\mathbf{a}}_{new}$ represents reshaping 2D matrix $\widetilde{\mathbf{A}}_{new}$ to 1D vector $\tilde{\mathbf{a}}_{new}$, and $\mathbf{b}_{new} \rightarrow \mathbf{B}_{new}$ is reshaping 1D vector $\mathbf{b}_{new}$ to 2D matrix $\mathbf{B}_{new}$. This reconstruction procedure is not limited to facial parts but can also be generally applied to other types of image synthesis.

We selected more than 10 subjects from the Cohn-Kanade database, each of which has around 70$\sim$140 images of different facial expressions, in addition to the image sequences we captured. For the image set of each subject, we randomly sampled one tenth of the images as the testing set, and the remaining images as the training set. We applied MCCA, CCA, and the standard linear least-squares regression (SR) approach to synthesize Mouth from Eye and vice versa on the testing set. We used 10 randomly selected training/testing combinations for reporting reconstruction errors. We observe that MCCA performs better than CCA and SR in reconstructing one facial part from another. Moreover, MCCA requires much fewer canonical factors to obtain better reconstruction results. We report the reconstruction results for six random selected subjects in Table 1, where the optimal average pixel errors (with standard deviation) and the corresponding dimensions of canonical factors used are reported. To clearly compare the three methods, we plot bar graphs of average pixel errors and the dimensions of the canonical factors used in MCCA/CCA in Fig. 4. Some reconstruction examples are shown in Fig. 5 (A supplementary video demonstration is available at *http://www.dcs.qmul.ac.uk/$\sim$cfshan/research/cca.html*).

Subject	Algorithm	Eye → Mouth		Mouth → Eye	
		Pixel Errors	Dims	Pixel Errors	Dims
(1)	2DCCA	11.2±2.0	11*6	8.8±1.2	2*23
	CCA	16.7±4.4	139	13.1±4.0	139
	SR	17.3±3.5	-	14.7±4.8	-
(2)	2DCCA	8.5±2.5	9*6	8.4±1.8	5*10
	CCA	13.0±6.0	119	10.7±3.6	119
	SR	12.4±5.3	-	10.7±3.2	-
(3)	2DCCA	13.4±5.5	15*3	10.0±3.3	28*1
	CCA	16.2±8.8	96	11.6±5.9	96
	SR	16.1±8.8	-	12.0±6.5	-
(4)	2DCCA	16.3±5.0	39*1	19.5±6.0	17*3
	CCA	24.5±9.8	96	25.4±18.3	96
	SR	23.7±7.6	-	26.1±18.8	-
(5)	2DCCA	9.9±1.8	14*2	10.5±2.5	22*2
	CCA	12.9±4.4	85	11.0±3.1	85
	SR	14.2±4.3	-	11.0±2.9	-
(6)	2DCCA	13.8±2.4	28*1	12.6±2.9	18*2
	CCA	17.2±8.5	77	15.7±6.2	77
	SR	15.1±4.9	-	13.9±6.1	-

Table 1: The reconstruction results for six subjects: the optimal average pixel errors (with standard deviation) of the three algorithms, and the corresponding dimensions of canonical factors used in MCCA and CCA.

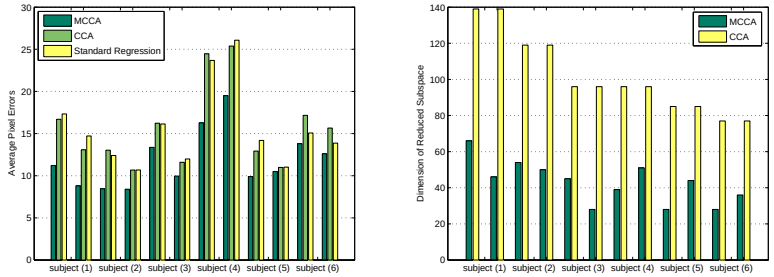


Figure 4: (*left*) Reconstruction errors of the three algorithms; (*right*) Dimensions of canonical factors used in MCCA and CCA. (Two groups bars for each subject: the left is 'Eye → Mouth' and the right is 'Mouth → Eye'.)

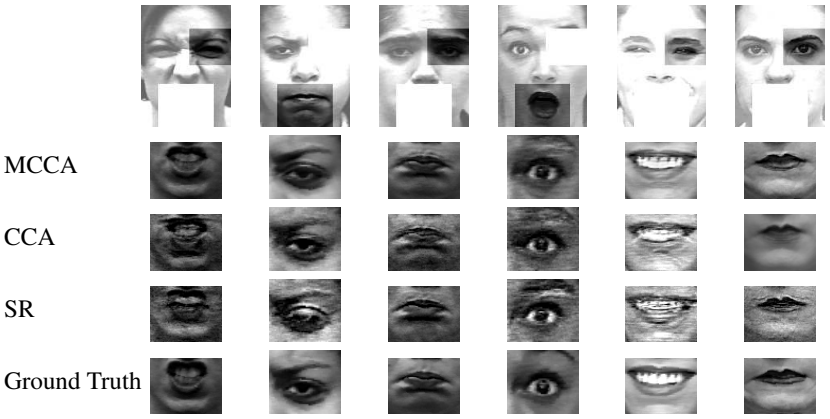


Figure 5: Some examples of facial parts synthesis using MCCA, CCA, and SR.

It is compelling that MCCA outperforms CCA and SR consistently in facial parts synthesis. Crucially, as observed in Fig. 4, the dimension of canonical factors needed in MCCA is always less than 50% of that of CCA. So MCCA can describe correlations among facial parts with better accuracy using much less canonical factors. The superior performance of MCCA credits to its ability to preserve the intrinsic 2D spatial structure and capture the correlation store therein, and its robustness with limited number of training data. The strength of MCCA is also reflected by the average standard deviation. As shown in Table 1, MCCA always produces the smallest deviation, which suggests that MCCA is much more robust. Compared to SR, where the full-rank regression matrix has to be estimated from a limited number of noisy training images, the MCCA based reduced-rank regression provides more reliable parameter estimates by taking advantage of correlations between the image sets, leading to better accuracy and robustness.

4.2 Facial Expression Recognition

We also conducted facial expression recognition experiments based on correlations between Mouth and Eye. The basic idea is that these two parts have distinctive correlations for different expressions, so the correlations modeled by MCCA should provide discriminant information for expression classification. Given image sets of different facial expressions $I_1,\ldots,I_c$ (c is the number of classes), we derive the leading factor pairs $(\mathbf{V}_a^i,\mathbf{W}_a^i,\mathbf{V}_b^i,\mathbf{W}_b^i), i=1\ldots c$ of parts Mouth (denoted by $\mathbf{B}$) and Eye (denoted by $\mathbf{A}$) for each class using MCCA. We then compute the regression parameters for reconstructing $\mathbf{B}$ from $\mathbf{A}$ in the reduced correlation space in the training set. Given a test image $\mathbf{I}_{new}$ of an unknown class, we map its Eye $\mathbf{A}_{new}$ and Mouth $\mathbf{B}_{new}$ to the reduced correlation space of class i as $\widetilde{\mathbf{A}}_i=(\mathbf{V}_a^i)^T\mathbf{A}_{new}\mathbf{W}_a^i$ and $\widetilde{\mathbf{B}}_i=(\mathbf{V}_b^i)^T\mathbf{B}_{new}\mathbf{W}_b^i$, and then calculate the error $err(i)$ of reconstructing $\widetilde{\mathbf{B}}_i$ from $\widetilde{\mathbf{A}}_i$ with the regression parameters of this correlation space. After computing the reconstruction error of each class $err(i), i=1\ldots c$, we classify the test image as the class having the smallest reconstruction error

$$\hat{i}=\arg\min_i err(i) \tag{12}$$

For our experiments, we selected 732 image of basic emotions (Anger, Disgust, Joy, and Surprise) from the Cohn-Kanade database. The sequences come from 96 subjects, with 1 to 4 emotions per subject. We first considered a 2-class (Joy and Surprise) recognition problem, then included Anger for a 3-class problem, and finally considered four expressions for classification (incrementally making the recognition task harder). To evaluate generalization performance, a 10-fold Cross-Validation testing scheme was adopted. The recognition results using MCCA and CCA are reported in Table 2. We can observe that expressions can be better classified using MCCA, demonstrating again that MCCA outperform CCA in capturing correlations in facial parts. It is also evident that by modeling correlations between only two facial parts, the recognition accuracy degrades quickly for multi-class recognition. By considering correlations of multiple facial parts, we should be able to improve these recognition results.

	2-Class	3-Class	4-Class
MCCA	96.1±3.6	80.8±6.4	67.9±4.8
CCA	63.2±10.5	55.6±7.8	48.7±6.7

Table 2: Facial expression recognition based on correlations of Mouth and Eye modeled by MCCA and CCA.

5 Conclusions

Experimental results have shown that the proposed MCCA can better model correlations among image data with much fewer canonical factors. The underlying reason is that MCCA is able to preserve and utilize the intrinsic 2D spatial structure in image data. MCCA is still a linear technique, however, so it cannot effectively deal with higher-order statistics among image data. Our future work will focus on formulating nonlinear MCCA.

References

[1] M. Borga. *Learning Multidimensional Signal Processing*. PhD thesis, Linkoping University, SE-581 83 Linkoping, Sweden, 1998. Dissertation No 531.

[2] I. Cohen, N. Sebe, A. Garg, L. Chen, and T. S. Huang. Facial expression recognition from video sequences: Temporal and static modeling. *CVIU*, 91:160–187, 2003.

[3] G. Dai and D.-Y. Yeung. Tensor embedding methods. In *National Conference on Artificial Intelligence (AAAI)*, pages 330–335, 2006.

[4] G. Donato, M. Bartlett, J. Hager, P. Ekman, and T. Sejnowski. Classifying facial actions. *IEEE PAMI*, 21(10):974–989, October 1999.

[5] R. Donner, M. Reiter, G. Langs, P. Peloscheck, and H. Bischof. Fast active appearance model search using canonical correlation analysis. *IEEE PAMI*, 28(10):1690–1694, October 2006.

[6] G. H. Golub and H. Zha. The canonical correlations of matrix pairs and their numerical computation. Technical report, 1992.

[7] D. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical correlation analysis; an overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, 2004.

[8] H. Hotelling. Relations between two sets of variates. *Biometrika*, 8:321–377, 1936.

[9] T. Kanade, J.F. Cohn, and Y. Tian. Comprehensive database for facial expression analysis. In *IEEE FG*, 2000.

[10] T-K. Kim, J. Kittler, and R. Cipolla. Learning discriminative canonical correlations for object recognition with image sets. In *ECCV*, pages 251–262, 2006.

[11] G. Littlewort, M. Bartlett, I. Fasel, J. Susskind, and J. Movellan. Dynamics of facial expression extracted automatically from video. *Image and Vision Computing*, 24(6):615–625, June 2006.

[12] T. Melzer, M. Reiter, and H. Bischof. Appearance models based on kernel canonical correlation analysis. *Pattern Recognition*, 39(9):1961–1973, 2003.

[13] M. Pantic and L. Rothkrantz. Automatic analysis of facial expressions: the state of art. *IEEE PAMI*, 22(12):1424–1445, 2000.

[14] Y. Tian, T. Kanade, and J. Cohn. Recognizing action units for facial expression analysis. *IEEE PAMI*, 23(2):97–115, February 2001.

[15] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H. Zhang. Multilinear discriminant analysis for face recognition. *IEEE Transactions on Image Processing*, 16(1):212–220, January 2007.

[16] J. Yang, A. F. Zhang, D. Frangi, and J. Yang. Two dimensional pca: A new approach to appearance-based face representation and recognition. *IEEE PAMI*, 26(1):131–137, 2004.

[17] J. Ye. Generalized low rank approximations of matrices. *Machine Learning*, 61:167–191, November 2005.

[18] C. Zou, N. Sun, Z. Ji, and L. Zhao. 2dcca: A novel method for small sample size face recognition. In *IEEE Workshop on Application of Computer Vision*, 2007.

Facial Emotions and Emotion Intensity Levels Classification and Classification Evaluation

Ing. Marián Beszédeš
Faculty of Electrical Engineering
and Information Technology
Slovak University of Technology
Bratislava, 81219, SK
beszedes@ktl.elf.stuba.sk

Dr. Phil Culverhouse
School of Computing,
Communications and Electronics
University of Plymouth
Plymouth, PL48AA, UK
P.Culverhouse@plymouth.ac.uk

Abstract

In this paper we analyze the problem of human facial emotion and emotion intensity levels recognition and resulting classification accuracy evaluation. Final testing set classification accuracy value is usually taken as a quantifier of method quality. However, this value is often strongly affected by the testing set parameters such as number, age and gender of subjects or intensity of their emotions etc. In this work we propose a different classifier evaluation methodology that uses the human visual system as a reference point. We employed active appearance models and support vector machines for facial emotion classification. Our SVM classifier gave slightly more consistent labels to emotion categories for images than human subjects, while humans were more consistent at identifying emotion intensity level than SVM.

1 Introduction

Automatic facial expression analysis has become an interesting research area since the early 90's because of many potential applications in areas like human-computer interfaces, face appearance synthesis, image retrieval and human emotion analysis.

Ekman and Friesen [6] has postulated six primary emotions which seem to be universal across human ethnicities and cultures. This primary emotions are commonly referred as *basic emotions* and comprise happiness, anger, surprise, disgust, fear and sadness. Despite many questions that arise around this study [12] (are the basic emotions indeed universal? Does a facial expression have a strong relation with an actual emotion state ?) it is still widely used in computer vision community.

While the human capability for face detection is very robust, the same has not been proven for facial expressions interpretation. According to Bassili [1], a person trained to classify faces can recognize six basic emotions with an average accuracy 87%. This accuracy ratio can change due to several reasons like the familiarity with the face, general experience with different types of expression, the facial emotion intensity level or even subjects or recognizing person race. For example the smiling face with low intensity can be easily misinterpreted as a neutral facial expression. Most current works related to automatic facial emotion recognition deal just with facial emotions having the highest

intensity. The classification of emotion intensity levels is in most of the cases omitted too. Surveys with more details related to analysis of facial expressions and emotions can be found in [7] or [11].

Facial emotion recognition in still images is connected with three basic tasks: 1) face detection, 2) facial expression data extraction and finally, 3) facial classification. There have been many techniques for face detection invented in last 15 years. For example the Viola-Jones face detector based on Haar feature classification and AdaBoost algorithm [13] combined with accurate skin color detector [2] can solve this complex task in real-time with very good classification accuracy. Fiducial grid and Gabor wavelets [9] or PCA analysis of random patches [10] are just examples of many other methods that were employed for the 2nd task of facial expression data extraction. Active Appearance Models (AAM) method is another well known method for accurate facial feature detection and facial emotion features extraction is approach [5],[8]. It can be relatively easily implemented, facial features are detected very fast and precisely and it yields good results on difficult and noisy data. Finally, the 3rd task of extracted facial data classification can be resolved using many types of classifiers. PCA, LDA and Mahalonobis distance [5],[9], neural network based classifiers [10] or SVM classifiers [8] methods were utilized for this aim.

The above methods are usually assessed on testing sets after all training procedures. The value that stands for portion of correctly classified samples from testing set is taken then as *final classification accuracy* κ and quantifier that tells about qualities of used method. However, κ is often strongly affected by the testing set parameters. Number, age, race, and gender of subjects; number, type and intensity of emotions; quality of images, lighting conditions; all these factors can impact the classification results. For example, in [8] Liebelt recognized 7 facial emotions with $\kappa = 71.3\%$ and no details about testing set parameters were given. Lyons used set of 193 images showing 9 Japanese females and 7 facial emotions for training in [9]. $\kappa = 75\%$ was achieved for novel subjects but no details about this kind of testing set were given again. 84 Ekmans' facial emotion photos with 12 subjects and 7 emotions were used in the work of Padget [10]. They trained their classifier on images of 11 subjects and tested on the images of the 12th subject. By changing the training and testing set they get average $\kappa = 86\%$.

There is no commonly used database of facial images with varying emotions as can be seen from previously mentioned works. One of the reasons of this lack is that researchers usually need images of different quality, lighting conditions, contrast, bit depth of colors etc. If we want to evaluate the classification results without considerations of testing set parameters, a reference classifier is needed: such as the human visual system.

In this paper we describe a classification system based on Active Appearance Models (AAM) method which is used for facial emotion features extraction and Support Vector Machines (SVM) method, which is used for classifiers training. An outline of the theoretical basis of the AAM model and the way we use it for proper feature extraction is presented. Then, important SVM classifier attributes are discussed as well as procedures suitable for its training. Our machine methods are referenced to a study that used human participants. The experimental methodology is presented and results of statistical analysis discussed in the context of the machine performance data.

2 Active Appearance Models

AAM learn characteristics of objects during a training phase by building a compact statistical model representing shape and texture variation of objects. As it is described in [4], the concept AAM is based on the idea of combining both shape and texture information of the objects to be modeled. The appearance model is built based on a set of N labeled images, where n key landmark points are marked on each example object and form so called shape vectors. The shape vectors $\mathbf{x}^i = (x_1^i, \ldots, x_n^i; y_1^i, \ldots, y_n^i)^T, i = 1 \ldots N$ are aligned using Procrustes analysis and the labeled images are warped to the mean shape $\overline{\mathbf{x}}$ and normalized, yielding the texture vectors $\mathbf{g}^i$. By applying principal component analysis (PCA) to the normalized data, linear models are obtained for both shape, $\mathbf{x} = \overline{\mathbf{x}} + \mathbf{P}_s \mathbf{b}_s$, and texture, $\mathbf{g} = \overline{\mathbf{g}} + \mathbf{P}_g \mathbf{b}_g$, where $\overline{\mathbf{x}}$, $\overline{\mathbf{g}}$ are the mean vectors, $\mathbf{P}_s$, $\mathbf{P}_g$ are sets of orthogonal modes of variation (the eigenvectors resulting from PCA), and $\mathbf{b}_s$, $\mathbf{b}_g$ are sets of model parameters.

A given object can thus be described by $\mathbf{b}_s$ and $\mathbf{b}_g$. As $\mathbf{P}_s$, $\mathbf{P}_g$ may still be correlated, one more PCA is applied to them. This yields the final combined linear model $\mathbf{b} = \mathbf{P}_c \mathbf{c}$, where $\mathbf{P}_c = (\mathbf{P}_{cs} \mathbf{P}_{cg})^T$.

The goal of AAM search is to find the model parameters that can generate a synthetic image as close as possible to given input and to use resulting AAM parameters for interpretation [4]. There has been developed several AAM search strategies that are precise enough to detect facial features. However, in our case we decided to use manually placed landmark points for shape definition, because AAM search can sometimes result in false detections which would decrease final emotion and emotion intensity level classification accuracy.

An AAM was employed for extraction of facial expression features of three types: shape parameters vector $\mathbf{x}$, texture parameters vector $\mathbf{g}$ and combined parameters vector $\mathbf{c}$. The amount of detail, or variance Ψ, contained in AAM model can be controlled by changing the number of modes of variation (number of eigenvectors) contained in $\mathbf{P}_s$, $\mathbf{P}_g$, $\mathbf{P}_c$. This number of variation modes determines the size of $\mathbf{x}$, $\mathbf{g}$ and $\mathbf{c}$.

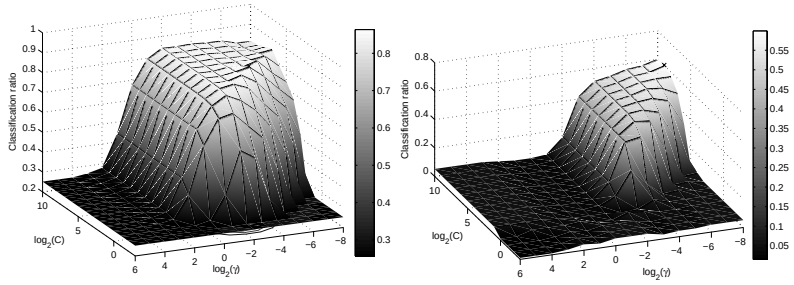
3 Support Vector Machine classifier

SVM classifiers with Gaussian RBF kernels were used for the task of emotion and emotion level classification. SVM classifiers are well known for their good generalization properties even in cases of high-dimensional nonlinear separable classification tasks. The RBF kernel was chosen because it has fewer adjustable parameters than any other commonly used kernel and has less numerical difficulties [3].

SVM classifiers can work well only when optimal values for its parameters are set. SVM classifier with Gaussian RBF kernel has two parameters: 1) the SVM regularization constant $C > 0$ and 2) γ parameter related to Gaussian RBF kernel function $k(\mathbf{x}, \mathbf{y}) = exp(\frac{-\|\mathbf{x}-\mathbf{y}\|^2}{\gamma})$. V-fold cross-validation and Grid-search algorithm can be utilized for this purpose. In v-fold cross-validation, the training set is first split into v subsets of equal size. Sequentially the classifier with defined learning parameters is trained v times, where in the i-th iteration ($i = 1, \ldots$,v) it is trained on all subsets except the i-th one. The classification error E_i is computed for the i-th subset. This procedure calculates v values of classification error where the average value of them $E = \frac{1}{v} \sum_{i=1}^{v} E_i$ is a rather good estimate of the classifier generalization error. The precision of classifier generalization error estimation

can be improved when several v-fold cross-validations (with different initial training set split) is performed and their results are averaged. The same approach can help to improve training accuracy when training data has small number of samples.

Grid search basically tries to estimate the generalization error of classifier using k ($k = mn$) different pairs of (C, γ) ($\{(C,\gamma)|C \in \{C_1, C_2, \ldots, C_m\}; \gamma \in \{\gamma_1, \gamma_2, \ldots, \gamma_n\}\}$). The pair with lowest generalization error is selected as an optimal. It was found [3] that trying exponentially growing sequences of C and γ is a practical method to identify good parameter values.



(a) (b)

Figure 1: Visualization of v-fold cross-validation results. Grid-search approach ($C = \{2^{-3}, 2^{-2}, \ldots, 2^{10}\}$, $\gamma = \{2^{-8}, 2^{-7}, \ldots, 2^{6}\}$) was used to find the optimal values (the black crosses) for the RBF SVM classifier parameters. (a) Emotion classifier, (b) Happiness intensity levels classifier

4 Experiments

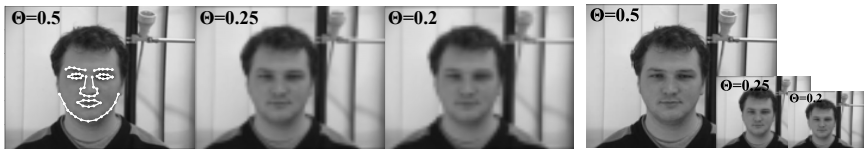
Facial emotion images were obtained from FG-NET facial emotion database [14] containing face image sequences showing a number of Caucasian subjects performing the six different basic emotions defined by Eckman [6]. One of the underlying paradigms of this database is to let the observed people react as naturally as possible (real emotions were elicited by showing video clips or still images after a short introduction phase). We decided to exclude emotions of sadness and fear because according to our observations they do not reach the same level of intensity and are not as distinctive as the other basic emotions. Images of selected emotions comprising anger, happiness, disgust and surprise (3 emotion levels for each emotion) plus neutral expression images of 12 individuals were selected from the FG-NET database and used during the analysis (see Figure 2(a)). The particular emotion intensity level images were selected from the image sequences on the basis of subjective selection. Additionally, all 192 faces were manually labeled with 58 landmark points defining face shapes. The placement of landmark points can be seen in Figure 2(b) (first image from left).

One of our aims was to analyze how the image quality affects the recognition accuracy. For this purpose we used source images (size 320×240) to generate images with reduced size. Size reduction factors $\Theta = \{0.5, 0.25, 0.2\}$ and bilinear interpolation method were used for generating of images with reduced size (see Figure 2(c)). In the case of psycho-

logical experiment the images were too small so they were enlarged to size 800×600 (see Figure 2(b)).

Images of 8 subjects (4 males, 4 females) served as training samples for AAM and SVM classifiers. The remaining images (2 males, 2 females) form the testing set for SVM classifier as well as for psychological experiment.

(a)



(b) (c)

Figure 2: (a) Examples of expression emotion images. 4 individuals performing emotion of disgust, surprise, anger and happiness with 4 emotion level intensities (neutral face expression stands for zero emotion intensity level). Examples of image quality degradation for psychological experiment (b) and size reduction for SVM training (c). Landmark points defining face shape can be seen in the most left image of (b).

.

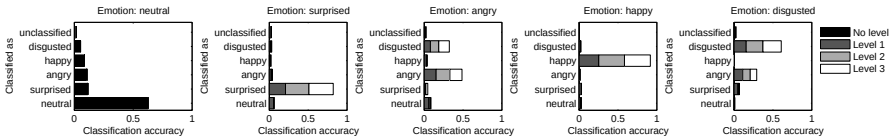
4.1 Psychological Experiment

Psychological experiment which should evaluate the ability of human respondents to recognize facial emotions and emotion intensity levels was prepared. Each trial of experiment consists of three parts - slides. In the first slide, an unmasked facial emotion image is presented and participant has 5 sec. to classify one of 5 emotions using the computer mouse to make the choice. Another slide with emotion intensity levels choices is shown for the same image. Again, the respondent has 5 sec. to choose the right choice. If the participant does not meet the time limit a "nothing" string value is saved as the response in both cases. Furthermore, if participant classify emotion in the first slide as neutral then the chosen level response in the second slide is not recorded. The last slide has no visual content and lasts for 100 ms, it serves as separator between trials.

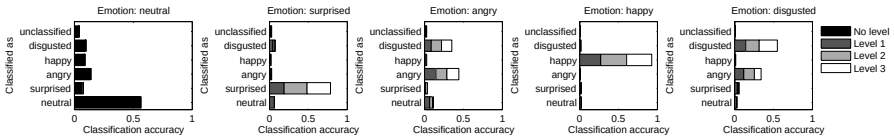
In each experimental session, participants perform 192 trials shown in random order, which followed a 4 individual $\times$ 4 emotions $\times$ 4 levels (neutral expression which serves as zero emotion intensity level is added to mentioned three emotion intensity levels) $\times$ 3 different image quality design. Twenty-four volunteers (12 males, 12 females) with normal or corrected-to-normal vision participated in this study. They had no prior knowledge about the subjects. They were instructed in detail and images of 2 individuals (1 male, 1 female, all emotions and emotion levels, size reduction factor $\Theta = 0.5$) with correct

information about their emotional state were shown to them before the experiment.

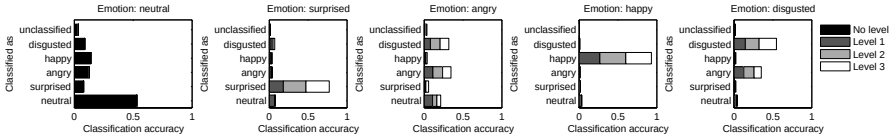
It can be seen from the analysis of results shown in Figure 3 that positive emotions comprising happiness and surprise are recognized with very good accuracy, and almost independently on image reduction. Negative emotions including anger and disgust are often mistaken and also neutral expression is often misclassified. It can be also seen how the accuracy decrease witch increasing Θ. The analysis of results for emotion intensity level classification can be found in Figure 4. It can be summarized that neighboring levels were often mistaken and that highest level was chosen the fewest times. The final classification accuracy can be found in Table 1.



(a) $\Theta = 0.5$



(b) $\Theta = 0.25$



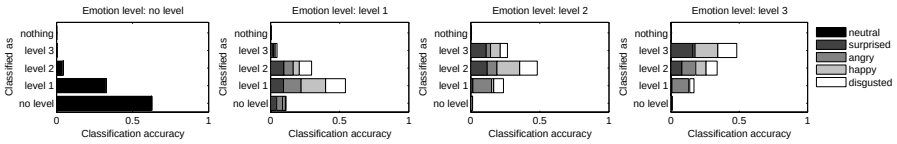
(c) $\Theta = 0.2$

Figure 3: Analysis of emotion classification results retrieved from psychological experiment. The histograms show what were the respondent's choices in comparison with the original emotion labels and bar colors shows how the original emotion levels affected the respondents choices.

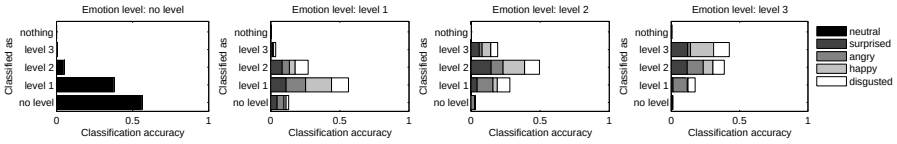
4.2 SVM Classifications

Thirty-six types (3 sizes of source facial images ($\Theta = \{0.5, 0.25, 0.2\}$) $\times$ 3 AAM parameter vector types (shape, texture, combined) $\times$ 4 values for variance retained in AAM models ($\Psi = \{0.93, 0.95, 0.97, 0.99\}$) of training data were tested for SVM classifier training. Grid-search approach ($C = \{2^{-3}, 2^{-2}, \ldots, 2^{10}\}$, $\gamma = \{2^{-8}, 2^{-7}, \ldots, 2^{6}\}$) was used to find the optimal values for the RBF SVM classifier parameters. Examples of grid-search classification results can be seen in Figure 1(a) (emotion classifier, results averaged over 4 tenfold cross validations (combined parameters vectors, $\Psi = 0.93$, $\Theta = 0.2$)) and 1(b) (happiness intensity level classifier, results averaged over 4 sixfold cross validations (shape parameters vectors, $\Psi = 0.93$, $\Theta = 0.2$)).

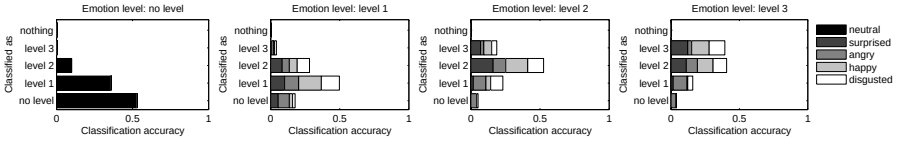
Five classifiers (1 classifier of emotions (128 training samples), 4 emotion intensity



(a) $\Theta = 0.5$



(b) $\Theta = 0.25$



(c) $\Theta = 0.2$

Figure 4: Analysis of emotion intensity level classification results retrieved from psychological experiment.

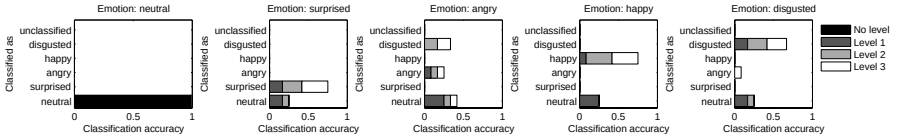
	SVM		**Respond.**	
Θ	**Emotions**	**Levels**	**Emotions**	**Levels**
0.5	**70.31%**	50.00%	68.75%	**53.26%**
0.25	**65.63%**	45.31%	64.71%	**51.11%**
0.2	**64.06%**	43.75%	61.85%	**48.70%**

Table 1: Final classification accuracy.

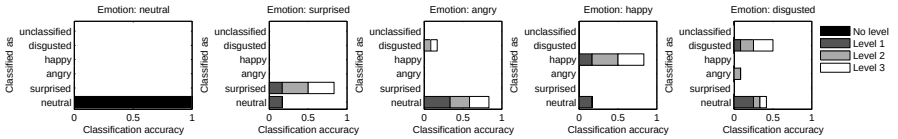
levels classifiers (32 training samples)) were trained using data type and SVM parameters that achieved best classification accuracy. Analysis of SVM classification results can be seen in Figure 5 and 6. The neutral emotion and "no level" choice are preferred by the classifiers because neutral images are presented with the highest number of samples in the training set. This can be seen especially in the case of negative emotions that are not as visually distinctive as positive emotions and in the case of first emotion intensity level.

5 Discussion and Conclusions

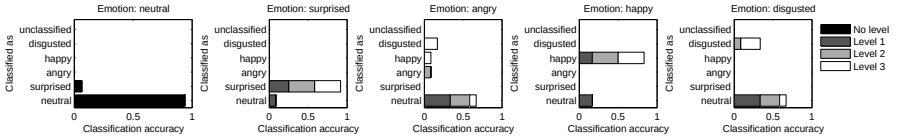
It can be seen in the Table 1 that SVM classifier outperforms the average classification accuracy of human respondents in the case of emotion classification for all image sizes. However, the SVM failed in the second case of emotion level intensity classification. This failure is probably caused by the small training set size. Moreover, the decrease of image sizes reduces the classification accuracy of both SVM classifier and the average accuracy



(a) $\Theta = 0.5$

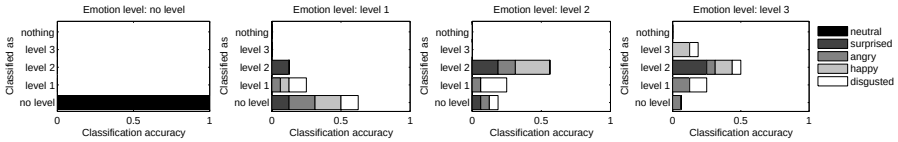


(b) $\Theta = 0.25$



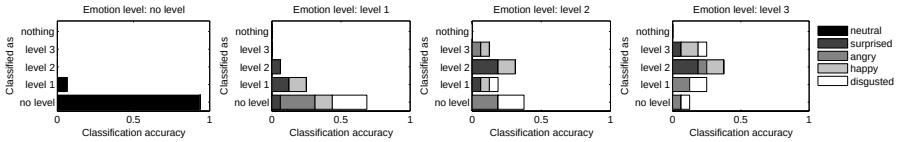
(c) $\Theta = 0.2$

Figure 5: SVM classifier emotion classification analysis.



(a) $\Theta = 0.5$

(b) $\Theta = 0.25$



(c) $\Theta = 0.2$

Figure 6: SVM classifier emotion intensity level classification analysis

.

of human respondents. This is caused by the loss of visual facial details that can be used for differentiation between emotions especially in the case of lower intensity levels.

				SVM				Respondents			
Emotion	*N*	*P*	Scale	*TP*	*FP*	*Acc*	*Prc*	*TP*	*FP*	*Acc*	*Prc*
neutral	48	16	0.5	16	14	0.78	0.53	10	2	**0.87**	**0.82**
			0.25	16	19	0.70	0.46	9	3	**0.85**	**0.77**
			0.2	15	19	0.69	0.44	9	4	**0.82**	**0.66**
surprised	52	12	0.5	9	0	**0.95**	**1.00**	10	4	0.91	0.74
			0.25	10	0	**0.97**	**1.00**	9	3	0.92	0.78
			0.2	11	1	**0.97**	**0.92**	9	2	0.92	0.80
angry	52	12	0.5	3	1	**0.84**	**0.75**	6	6	0.81	0.50
			0.25	1	0	**0.83**	**1.00**	5	7	0.79	0.44
			0.2	1	0	**0.83**	**1.00**	4	7	0.77	0.38
happy	52	12	0.5	9	0	**0.95**	**1.00**	11	2	0.95	0.84
			0.25	10	0	**0.97**	**1.00**	11	2	0.95	0.84
			0.2	10	1	**0.95**	**0.91**	11	3	0.93	0.77
disgusted	52	12	0.5	8	4	**0.88**	**0.67**	7	5	0.84	0.58
			0.25	6	2	**0.88**	**0.75**	7	7	0.81	0.49
			0.2	4	2	**0.84**	**0.67**	7	6	0.82	0.52
Level 1	32	16	0.5	4	8	0.58	0.33	9	7	**0.71**	**0.57**
			0.25	5	8	0.60	0.38	9	7	**0.70**	**0.55**
			0.2	4	7	0.60	0.36	8	6	**0.70**	**0.56**
Level 2	32	16	0.5	9	10	**0.65**	**0.47**	8	10	0.62	0.43
			0.25	3	5	**0.63**	**0.38**	8	11	0.61	0.43
			0.2	5	7	**0.63**	0.42	8	11	0.61	**0.43**
Level 3	32	16	0.5	3	0	**0.73**	**1.00**	8	5	0.72	0.61
			0.25	5	3	0.71	0.63	7	4	**0.73**	**0.65**
			0.2	4	2	0.71	**0.67**	6	4	**0.72**	0.63

Table 2: Analysis of ROC space characteristics. *Acc* - Accuracy, *Prc* - Precision, *N*/*P* - counts of negative/positive samples in testing set, *TP*/*FP* - counts of true positives / false positives classifications. $Acc = (TP + (N - FP))/(P + N)$; $Prc = (TP)/(TP + FP)$

The receiver operating characteristic space analysis for all source image sizes can be found in Table 2. In this case we treat the classification results as results retrieved from binary classifiers for individual emotions or emotion levels. We utilized Accuracy and Precision ROC space characteristics to compare average classification results of human respondents and classification results of SVM. The results highlight the variability innate in the human participant's responses to emotional cues. The emotional levels of image set were labeled by one person, who placed its interpretation on emotional content. It is therefore inaccurate to say that people are worse or better than the SVM in recognizing emotional cues in the case of emotional levels. It is better to say that SVM is more consistent than people in this study at labeling the gross class categories, with the exception of neutral emotion. Conversely, people were more consistent with emotional level identification than the SVM.

We have proposed an alternate and general way of classifier evaluation of emotional cues. This kind of evaluation is independent of testing data set structure and complexity. It can help to get classification quality quantifiers that can be compared with other methods

in more a human-centred and reliable way.

References

[1] J. N. Bassili. Facial motion in the perception of faces and of emotional expression. *Journal of Experimental Psychology Human Perception and Performance*, 4:373–379, 1978.

[2] M. Beszedes and M. Oravec. Adaboost algorithm used for skin color detection. *29th International Conference Telecommunications and Signal Processing TSP-2006 Brno, Czech Republic*, pages 96–99, Sept 2006.

[3] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[4] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

[5] G.J. Edwards, T.F. Cootes, and C.J. Taylor. Face recognition using active appearance models. *Proceedings of the European Conference on Computer Vision*, 2:581–695, 1998.

[6] P. Ekman and W. V. Friesen. Constants across cultures in the face and emotion. *Journal of Personality and Social Psychology*, 17, 1971.

[7] B. Fasel and J. Luettin. Automatic facial expression analysis: A survey. *Pattern Recognition*, 36(1):259–275, 2003.

[8] J. Liebelt, J. Xiao, and J. Yang. Robust aam fitting by fusion of images and disparity data. *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Volume 2*, pages 2483–2490, 2006.

[9] M.J. Lyons, J. Budynek, and S. Akamatsu. Automatic classification of single facial images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(12):1357–1362, 1999.

[10] C. Padgett and G. Cottrell. Representing face images for emotion classification. *Advances in Neural Information Processing Systems*, 9:894–900, 1997.

[11] M. Pantic and L.J.M. Rothkrantz. Automatic analysis of facial expressions: the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, 2000.

[12] J.A. Russell. Is there universal recognition of emotion from facial expression? a review of the cross-cultural studies. *Psychological Bulletin*, 115(1):102–141, 1994.

[13] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Proc. CVPR*, 1:511–518, 2001.

[14] F. Wallhoff. Facial expressions and emotion database. `http://www.mmk.ei.tum.de/~waf/fgnet/feedtum.html`, 2006.

Segmented AAMs Improve Person-Independent Face Fitting

Julien Peyras[1] Adrien Bartoli[2] Hugo Mercier[3] Patrice Dalle[3]

[1] Dipartimento di Scienze dell'Informazione, Milano, Italy
[2] LASMEA, Clermont-Ferrand, France
[3] IRIT, Toulouse, France

{Peyras,Mercier,Dalle}@irit.fr Adrien.Bartoli@gmail.com

Abstract

An Active Appearance Model (AAM) is a variable shape and appearance model built from annotated training images. It has been largely used to synthesize or fit face images. Person-independent face AAM fitting is a challenging open issue. For standard AAMs, fitting a face image for an individual which is not in the training set is often limited in accuracy, thereby restricting the range of application.

As a first contribution, we show that the limitation mainly comes from the inability of the AAM appearance counterpart to generalize, *i.e.* to accurately generate previously unseen visual data. As a second contribution, we propose an efficient person-independent face fitting framework based on what we call multi-level segmented AAMs. Each segment encodes a physically meaningful part of the face, such as an eye. A coarse-to-fine fitting strategy with a gradually increasing number of segments is used in order to ensure a large convergence basin.

Fitting accuracy is assessed by comparison with manual labelling statistics constructed from multiple data annotations. Experimental results support the claim that standard AAMs are well-adapted to person-specific fitting while segmented AAMs outperform the classical AAMs in a person-independent context in terms of accuracy, and ability to generate new faces.

1 Introduction

The Active Appearance Model (AAM) paradigm was introduced in 1998 by Cootes *et al.* [3] and since then it has had a great success. An AAM learns the shape and the appearance of a labelled set of images showing some class of objects. AAMs are widely used for face fitting, see *e.g.* [3, 8] and face synthesis, see *e.g.* [4]. Most of the applications – in the medical, psychological and linguistic fields, cognitive studies, expression transfer on an avatar, *etc.* – require highly accurate fitting. In other words, the AAM parameters must be recovered such that the synthesized image closely matches the input image.

Most of the previous work uses a single AAM modeling the face as a whole. Accurate fitting is achieved in a person-specific context. For instance, [8] uses AAMs for facial

deformation analysis. The standard AAM usually fails to achieve high accuracy for an image of a previously unseen face, *i.e.* for an individual not in the training set. Person-independent face fitting is however a very important problem since a training image set might not be available for an individual whose face needs to be accurately tracked in a video.

The closest work to ours is probably by Gross *et al.* [7]. They tackle the problem of constructing and fitting person-independent AAMs. They show that this is a difficult problem, even for frontal pose and neutral expression, and that the difficulties come from the inability of standard AAMs to generate new faces. A solution based on training, iteratively refitting the data with the AAM and re-training, is shown to improve the performances compared to traditional single step AAM training. Cristinacce *et al.* [5] recently proposed a paradigm called Constrained Local Model (CLM). It is shown to be effective at fitting a local face model based on measuring the image response around vertices and with a shape prior learnt from training images.

This paper tackles the important issue of person-independent face fitting with AAMs. We bring several statements and technical contributions:

- First, §3, we propose a means to assess fitting accuracy: the SSE (Statistical Shape Error). It is based on using several manual labellings of the input images by different users, from which gaussian statistics are computed for each label. The quality of an AAM fit is assessed by using the Mahalanobis distance with manual labelling statistics. This is an essential tool for the subsequent experimental analysis.

- Second, §4, we experimentally investigate the behavior of standard AAMs on unseen faces, and show that the lack of accuracy is mainly due to the inability of the appearance component to generate unseen faces. We state that standard AAMs are accurate in a person-specific context but not in a person-independent one.

- Third, §5, we show that segmented AAMs outperform standard ones in the person-independent context and achieve very accurate fitting, of the same order as the accuracy reached with manual labelling statistics. Segmented AAMs consist of several portions, each of which modeling a region of the face such as the mouth. Directly fitting each segment would reduce the convergence basin compared to fitting a standard AAM. As a remedy, we propose a coarse-to-fine fitting strategy which gradually splits a standard AAM into pre-defined segments. This *multi-level segmented AAM* we propose thus is able to generate new faces and can be effectively fit to images. Experimental results show that this outperforms the refitting solution of [7].

We give some background on AAMs below and our conclusions in §6.

2 Background on AAMs: Training and Fitting

An AAM combines two linear subspaces, one for the shape and one for the appearance, which are learnt from a previously labelled set of training images [3].

Principal Component Analysis (PCA) is applied on shape training data to retrieve a set of shape eigencomponents s_i expressing the shape model variation, and their associated eigenvalues proportional to the variance of the training data the s_i enclose. Four

extra components s_i^* are added to allow the 2D similarity transform, see [10]. Let $B_s = [s_1, \cdots, s_i, \cdots, s_1^*, \cdots, s_4^*]$ be the shape subspace basis. An instance of shape is defined as a linear expression: $s = B_s p_s$ with p_s the shape deformation parameters.

PCA is applied on the shape-corrected appearance data to retrieve a set of appearance eigencomponents A_i, allowing variations on the model appearance, and their associated eigenvalues proportional to the variance of the training data the A_i enclose. Two extra components A_0 for gain and A_I for bias are added, see [1]. Let $B_a = [A_1, \cdots, A_i, \cdots, A_0, A_I]$ be the appearance subspace basis. An instance of appearance is defined as a linear expression: $A = B_a p_a$ with p_a the appearance variation parameters.

Fitting an AAM consists to find the shape and appearance parameters that make it match the input image as best as possible. This is done by an iterative, nonlinear optimization process. We use the inverse compositional optimization scheme presented by Baker and Matthews in [10]. The Jacobian and Hessian matrices are derived analytically. Two versions of this algorithm were proposed and compared in [7]. Our implementation relies on the most accurate one called the *simultaneous inverse compositional algorithm*, originally described in [1].

We adapt this algorithm to our multi-level segmented AAM.

3 Assessing Fitting Accuracy

Fitting accuracy on unseen face images is generally assessed based on a single manual annotation of each image, considered as the absolute shape reference. The assumption behind this accuracy evaluation method is that the manual label is correct at the pixel level.

This assumption is often violated in practice: a vertex on a face image gets significantly different manual annotations, even from the same user. It is also incorrect to consider that one manual annotation is better than the others. It might also happen that a well performing automatic process is more accurate than manual labellers.

To address this improper accuracy assessment problem, Mercier *et al.* [11] suggest to annotate a face several times and build statistics for each vertex. It is then possible to set up a fitting error measure that takes the imprecision of manual annotation into account. The fitting accuracy score is given strong weight for those vertices that manual labellers have localised accurately, and light weight for badly localised vertices.

We use the multiple label data available from [11] to define the ground truth shape and the fitting error function. A set of $n_I = 40$ images were labelled $n_L = 10$ times each (labels describe the $n_V = 68$ vertices of the model mesh used in [10]). These frontal pose, neutral expression, homogeneous illumination, face images are extracted from the AR database [9]. Each image shows a different individual. A probability distribution is computed for each image i and vertex v, as the mean $\mu_{i,v}$ over its n_L labels $x_{i,v,l}$ and a (2×2) covariance matrix $\Sigma_{i,v}$ as:

$$\mu_{i,v} = \frac{1}{n_L} \sum_{l=1}^{n_L} x_{i,v,l} \quad \text{and} \quad \Sigma_{i,v} = \frac{1}{n_L - 1} \sum_{l=1}^{n_L} (x_{i,v,l} - \mu_{i,v})^T (x_{i,v,l} - \mu_{i,v}).$$

These define what we dub 'manual labelling statistics'. Figure 1 shows face images overlaid with their manual labelling statistics, with each vertex represented by an ellipse showing its mean position and uncertainty. This methodology is in contrast to [11] in which a

single covariance matrix is computed for each vertex over all the n_I images. We believe that keeping a single covariance matrix for each vertex in each image makes sense since the visibility conditions may substantially differ from one image to the others for the same vertex. We want to preserve this information in the statitics.

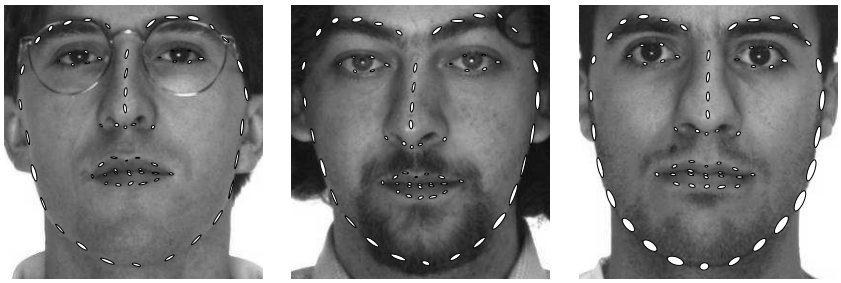


Figure 1: Faces number 1, 3 and 6 from the 40 faces that were annotated 10 times. Covariance ellipses represent the distribution of the 10 labels around mean vertices.

We propose the *Statistical Shape Error* (SSE) for a shape s on an image i that we define by the average of the Mahalanobis distances:

$$SSE_i(s) = \frac{1}{n_V} \sum_{v=1}^{n_V} \sqrt{(s_v - \mu_{i,v})^T \Sigma_{i,v}^{-1} (s_v - \mu_{i,v})}, \quad (1)$$

where s_v is the v-th vertex of shape s. The lower the SSE, the better the fitting accuracy. This error is strongly related to the negative log-likelihood of the parameters with respect to gaussian noise contamined labels. It scores automatic fits and can also be used to score manual fitting accuracy. In particular, we compute the SSE obtained by the 10 labellings on each image. From the 10 error scores on each image we retain the maximum and minimum scores, and compute the average score. This allows us to compare automatic fitting accuracy to manual fitting accuracy in §5.3, which gives a concrete idea of the accuracy that is reached.

4 Issues in Fitting AAMs to Unseen Faces

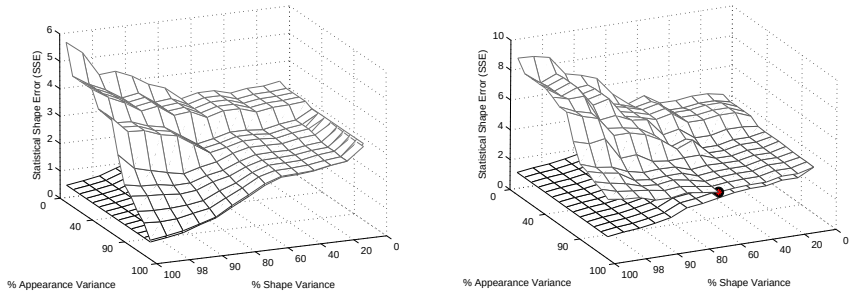
In the literature, AAMs are usually built by retaining 95 to 98% of the total shape and appearance variance contained in the training data without justifying this choice. Few works study the influence of the quantity of variance on the fitting performances. Gross *et al.* [7] recently investigated the effect of shape and appearance variance on the convergence of a fitting algorithm for unseen faces. They estimate the quantities of shape and appearance variance that maximize the number of successful trials. However, they do not explain why the convergence is limited for certain faces.

The experiment we report allows to highlight the fitting accuracy behavior for a range of shape and appearance variances. We identify the combination that maximizes the overall fitting accuracy on all trials and explain why this accuracy is limited and the fitting behaviour for various shape and appearance variance combinations. The experimental setup has similarities with the one in [7].

4.1 Fitting Seen Faces, *i.e.* Images in the Training Set

We use all the 40 images to train the AAM with different amounts of shape and appearance variance. We fit it to the 40 face images on turn. Each fitting trial lasts a number of iterations that allows to reach a clear final state, should it be convergence or divergence.

As in [7], we initialize the fitting process as close as possible to the optimal parameterization: we project the test face shape into the shape subspace to retrieve the initial shape parameters, and its appearance into the appearance subspace to retrieve the appearance parameters. In this way, we ensure that if the model diverges, it is not due to potential local minima but to the model inability to fit the test image. Figure 2 (a) shows the average SSE on all the 40 face images. The bottom curve shows the model SSE in its initial position, which is also the lowest error it can reach.



(a) Accuracy for **seen faces** (images in the training set)

(b) Accuracy for **unseen faces** (individuals not in the training set)

Figure 2: *Seen* and *unseen* contexts analysis. The bottom curves represent the initial SSE of equation (1) averaged over all the 40 images. The top curves show the average SSE after the algorithm has ran. Various amounts of shape and appearance variances are tested. The black dot in (b) represents the point of best average fitting accuracy on unseen faces that stands for 60% of the shape variance and 100% of the appearance variance.

We observe that for full appearance (100% of the variance retained), the fit remains in the best, initial position for any amount of shape variance retained. The characteristics of the full appearance AAM is that, up to appearance sampling artefacts, the test image can be completely reconstructed in appearance.

The second observation holds for any given fixed amount of shape variance: when less than 100% of the appearance variance is retained, the fitting accuracy decreases. The less the appearance variance, the worse the accuracy. For less than 100% of the training data, the AAM appearance space cannot totally reproduce the face appearance of the input image although this face is in the training set. This reluctant intensity discrepancy between the test image and the model causes the drop in the fitting accuracy. This test highlights the following property: the ability of the AAM appearance component to fully generate the face appearance of the test image is a necessary condition to obtain the best possible fitting accuracy. A natural question to answer is whether this also holds when the test image is not in the training set.

4.2 Fitting Unseen Faces, *i.e.* Individuals Not in the Training Set

This test is different from the one in §4.1 in that it is done in a leave-one-out manner: we train the AAM on 39 images and use the 40-th image as a test image of an unseen face. The test is performed for all the 40 face pictures, and for variable amounts of shape and appearance variance.

Figure 2 (b) shows the average SSE over the 40 face images. It is pretty similar to figure 2 (a) but a main difference is however observed. There is no combination of shape and appearance variances that makes the AAM remains into the initial, best position. Indeed, there is no junction between the SSE curves for the initial and fit curves. In contrast with the test on seen faces, even for full appearance AAMs, the fitting process shifts the AAM away from the initial solution. The AAM never remains on the best possible accuracy position, which results in limited accuracy capabilities.

4.3 Discussion

As an observation on the test for seen faces in §4.1, we saw that when the model can fully express the image in terms of appearance (the error in intensity between the model and the image are due to the model misplacement and/or non-optimal appearance parameterization). The fitting optimization process uses the error in intensity to iteratively update the model to a position where this error is minimized, and ideally equals zero. It is assumed that the model parameterization that minimizes the error in intensity correctly aligns the model to the face image. In practice, this is what happens when the model explicitly learnt the image it fits (and when the global minimum is reached). This explains the high fitting accuracy obtained in this context.

When the model appearance cannot fully express the face on the test image, the error in intensity due to this lack of expressivity is considered as being due to the model misplacement. The optimization process tunes the model parameters to minimize the residual error though it does not come from a misplacement. In this case, the minimum error usually does not correspond to the best placement of the model vertices. Indeed, the process bends the model in order to spread out the remaining error in intensity as much as it can to minimize the global error. This makes the model drift away from the sought after shape used as its initial position, *i.e.* fitting accuracy is spoiled. The more deformable the model the more the fitting process can bend it to further minimize error in intensity. For very high deformability the model can even diverge. In the same way for a given deformability (fixed amount of shape variance), the less the appearance variance, the less the model can express the test image data and the worse the fitting accuracy. In the case of fitting on seen faces, this happens when appearance variance is not fully retained (less than 100% of the variance is retained). In the case of fitting on unseen faces, a new face always presents visual aspects that are unknown from the model appearance component and the model always drifts away from the best possible position even when appearance is fully retained. As seen on the curves of figure 2 (b), the best overall accuracy for fitting on unseen faces is obtained for full appearance and 60% shape variance, making the model rigid enough not to bend too much, then minimizing the loss in fitting accuracy.

5 Segmented AAMs and Unseen Individuals

5.1 Motivations for Using Segmented AAMs

The AAM appearance space is unable to completely generate the appearance information. In other words, an unseen face added to the training set would bring new visual information. We saw that the limited ability to generalize the appearance component limits the fitting performance in terms of accuracy. One obvious solution to better generalize to any new face appearance would be to train the AAM on thousands of training images. This is difficult in practice for two reasons: first, this number of training data is hard to gather up, and second, this implies to retain a very high number of appearance components to explain as much of the variance as possible, which makes the optimization process computationally heavy and increases the possibility of getting stuck into local minima.

The solution we propose is to reduce the appearance space dimensionality. This makes more expressive the data coming from our reasonable size training set. To achieve a better fitting accuracy we rely on local models defined over a smaller face area. This approach is somehow similar to the concept of *segmented morphable models* briefly presented by Blanz and Vetter in [2].

5.2 Multi-Level Segmented AAMs and Coarse-to-Fine Fitting

The ability of local models to generalize their shape and appearance is better than for larger models. This makes them potentially more accurate for the same amount of training data. However, their reduced dimension penalizes their robustness to bad initialization: local models must be well initialized. To ensure this, we use a three stage coarse-to-fine strategy, illustrated on figure 3, where a global AAM is used to initialize intermediary AAMs, themselves used to initialize local AAMs.

Intermediary and local models represent a subgroup of the global model vertices. Models concerned with eyebrows also describe some extra vertices on lower eyebrows. A layer of *supporting points* is added to local and intermediary models in order to define visual gradients at 360° around all vertices.

The model is automatically initialized. We use a face and eye center detector available online[1] [6]. The rigid global model is transformed with a 2D similarity and is placed on the image such that its eye centers match the corresponding estimate given by the detector. From this initial position the fitting is launched until it converges.

Global model position is used to initialize each intermediary model: we keep the vertices the global model has in common with an intermediary model, and we find the intermediary model instance that best matches those vertices, as follows.

Let B_{succ} be the shape generating matrix of one intermediary model: the columns of B_{succ} are the n_C long deformation vectors s_i plus the four similarity transform vectors. Vector s_{curr} represents the vertex coordinates of the global model that are in common with the intermediary model. To this vector we add extra null coordinates for intermediary model vertices that are not in common with global model. s_{curr} thus becomes n_C long. We sort the coordinates in s_{curr} in a way such that they correspond to vertex coordinates defined in the vectors s_i. The instance of intermediary model that best matches its common vertices to those of the global model is found by solving the following optimization

[1] http://kolmogorov.sourceforge.net

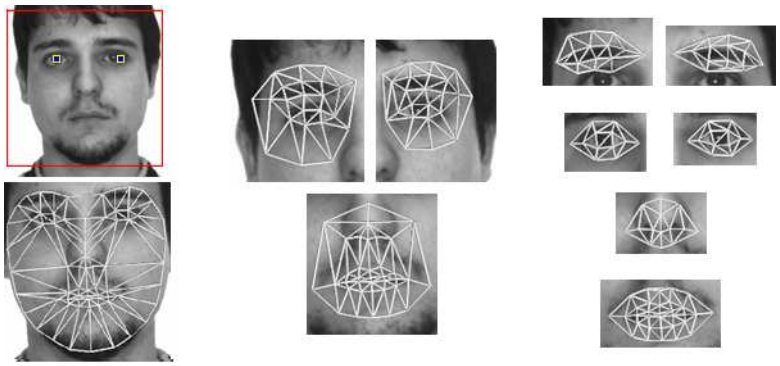


Figure 3: Illustration of the models used to fit the face. A global model is initialized with help of an eye center detector and is fitted on the image giving a first fitting result (left column). From this initialization a set of intermediary models are launched to further refine the fitting (center column). Eventually, the local models dedicated to each facial feature are launched to fit these features more accurately (right column).

problem:

$$\arg\min_{p} \sum_{c=1}^{n_C} Q(c)\left(s_{curr}(c) - B^c_{succ}p\right)^2, \tag{2}$$

where B^c_{succ} is the c^{th} row of matrix B_{succ}. Q is an n_C long vector of weights set to one for the coordinates of vertices that are common between the models, and to zero for the others. A closed form solution can be computed to find the optimal $p^\dagger$ (details are omitted due to lack of space):

$$p^\dagger = (K^T K)^{-1} K^T B^T_{succ} diag(Q) s_{curr}, \tag{3}$$

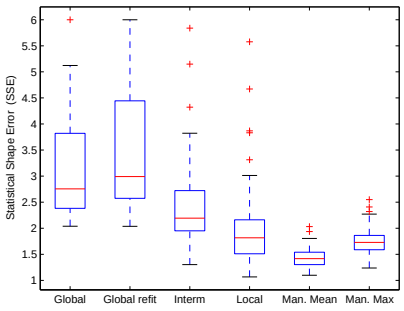
where $K = B^T_{succ} diag(Q) B_{succ}$ and $diag(Q)$ is a diagonal matrix, null everywhere excepted on diagonal where the Q vector coefficients are represented. The result $p^\dagger$ of this minimization can be used to instantiate the shape of the intermediary model: $s_{succ} = B_{succ} p^\dagger$. The process is applied to initialize all intermediary models that are then fitted to the image. Following the same strategy, we use (converged) intermediary model vertices to initialize local models that are in turn fitted to the image. Once each model is initialised in position, its appearance component is initialised by projection of the area underlying on the image onto the appearance subspace, in order to retrieve the initial appearance parameters.

5.3 Experimental Results

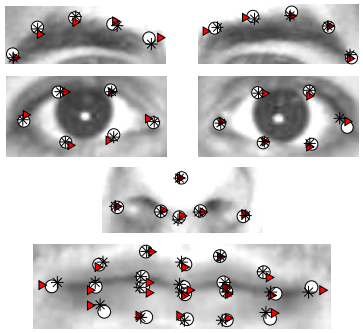
Fitting with intermediary and local models leads to improved fitting accuracy. On figure 4 (a), boxes of whiskers compare the SSE on 40 trials obtained respectively with global, global refitted, intermediary, local models and mean manual error as well as maximum manual error (see §3). The global refitted model is obtained by training the global model

onto refitted data: the used face data is learnt by an AAM retaining 99% variance of shape and appearance, and is fitted again on the same faces in order to increase the vertices correspondences among training data. Introduced in [7], this operation seems to improve the fitting results on unseen faces with respect to the results obtained when training the AAM on once-labelled data. Since we use multiply labelled data for training (the mean of 10 labels to define each vertex), their semantical position on the face is high and should naturally improve the correspondence among data.

A leave-one-out procedure is used to train and fit the global, global-refitted, intermediary and local models. Models are built with the shape and appearance variances that maximize their overall accuracy on unseen faces (*e.g.* 60% shape and 100% appearance for the global model). All intermediary models are gathered. The same is done for the local models. The SSE is computed using equation (1) only on vertices that are common between the models. We see the accuracy improvement allowed by intermediary and local models with respect to the global model, and we see that the accuracy is globally comparable to manual label accuracy evaluated with the statistics. The relative improvement obtained from global to local model fitting is 36% on average. The SSE obtained for refitted data is higher than for the global model trained with multiply labelled data. We believe that the higher semantical meaning obtained with label statistics is mainly responsible for the improvement. Indeed, these labels have higher semantical meaning since human labellers attempted several times to accurately set them into a given position on each face. The refitting process will displace once-labelled vertices to maximize their cohesion, but it is improbable that these new positions are semantically the very desired ones (although they might usually be improved). Multiply labelled data then constitute a maximum bound to accuracy, which explains the improved results obtained when we train an AAM with these data.



(a) Comparison results



(b) Example of fitting results

Figure 4: (a) Comparison between the 40 fitting scores obtained with a global model, a refitted global model, intermediary models, local models, and manual labellings, both maximum and average SSE. Local models often reach a SSE comparable to manual labellings. (b) Example of fitting results on face number 6. The circles represent the ground truth shape vertices (centers of the covariance ellipses), the triangles the vertices of the fitted global model (the SSE equals 2.46), and the stars represent the vertices of the fitted local models (the SSE equals 1.46).

6 Conclusion and Future Work

The AAM paradigm is often used without precisely understanding the influence of the *quantity and nature of training data* and of the *retained quantity of shape and appearance variance* on fitting performances. As a step towards such an understanding this work studies fitting accuracy on unseen frontal and neutral face data through the *Statistical Shape Error* we propose. We showed and explained the fitting accuracy limitations in this case. We propose a solution based on local models, namely the *multi-level segmented AAM*, that overcomes this limitation and reaches very high accuracy benchmarked by manual fitting accuracy with a large convergence basin. To summarize, standard and segmented AAMs are respectively well-adapted to person-specific and person independent face fitting.

We wish to extend these results to varying pose and expression: we will train one set of global, intermediary and local models for each possible pose and expression and set up a strategy to select the set that best suits for fitting the current face image.

References

[1] S. Baker, R. Gross, and I. Matthews. Lucas-Kanade 20 years on: A unifying framework: Part 3. Technical Report CMU-RI-TR-03-35, November 2003.

[2] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *Proceedings of* SIGGRAPH, 1999.

[3] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

[4] D. Cosker, D. Marshall, P. Rosin, and Y.A. Hicks. Speech driven facial animation using a hierarchical model. IEE *VISP*, 151(4):314–321, 2004.

[5] D. Cristinacce and T. F. Cootes. Feature detection and tracking with constrained local models. In *Proceedings of the British Machine Vision Conference*, 2006.

[6] I. Fasel, B. Fortenberry, and J. R. Movellan. Generative framework for real-time object detection and classification. *CVIU*, 98(1):182–210, April 2005.

[7] R. Gross, I. Matthews, and S. Baker. Generic vs. person specific active appearance models. *Image and Vision Computing*, 23(11):1080–1093, 2006.

[8] S. Lucey, I. Matthews, C. Hu, Z. Ambadar, F. de la Torre, and J. Cohn. AAM derived face representation for robust facial action recognition. *FGR*, 2006.

[9] A.M. Martinez and R. Benavente. The AR face database. Technical Report 24, CVC, June 1998.

[10] I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135–164, November 2004.

[11] H. Mercier, J. Peyras, and P. Dalle. Toward an efficient and accurate AAM fitting on appearance varying faces. *FGR*, 2006.

Gender Classification using Shape from Shading

Jing Wu, W.A.P. Smith, E.R. Hancock
Department of Computer Science, The University of York, York, YO10 5DD, UK.
{jwu, wsmith, erh}@cs.york.ac.uk

Abstract

The aim in this paper is to show how to use the 2.5D facial surface normals (needle-maps) recovered using shape from shading (SFS) to improve the performance of gender classification. We incorporate principal geodesic analysis (PGA) into SFS to guarantee the recovered needle-maps is a possible example defined by a statistical model. Because the recovered facial needle-maps satisfy data-closeness constraint, they not only give the facial shape information, but also combine the image intensity implicitly. Experiments show that this combination gives better gender classification performance than using facial shape or texture information alone.

1 Introduction

Humans are remarkably accurate determining the gender of a subject based on the appearance of the face alone. In fact, an accuracy as good as 96% can be achieved with the hair concealed, facial hair removed and no makeup [6]. In recent years, a lot of effort has been spent on the statistical features based [4], [5], [16], [15], [10] approaches for gender classification. Most of them are based on the 2D intensity information. Although studies [3] have shown that the gender is not only revealed by the 2D face textures, but also has close relationship with the 3D shapes of the human faces, only a few studies have investigated the 3D shapes in gender classification [10]. In [10], Lu and Chen exploit the range information of human faces for gender classification, and propose an integration scheme by combining the registered range and intensity images. The experiment results demonstrate that integrating the 3D range modality provides better classification accuracy than using the 2D intensity modality alone. However, the representation and computation is much more complex for 3D face shapes than 2D face textures. Moreover, due to the immaturity of the 3D sensors in the current market, some typical problems with range images including missing data near dark regions and spikes at the region with high reflectivity would deteriorate the classification.

In this paper, we present a statistical framework for gender classification based on the 2.5D facial needle-maps. The needle-map is a shape representation which can be acquired from 2D intensity images using shape from shading (SFS). Therefore it avoids the problems caused by the immaturity of the current 3D sensors, and provides the shape information from a fixed view point. In [14], a new iterative SFS method is proposed, which not only satisfies the data-closeness constraint (satisfies the image irradiance equation)[17], but also guarantees the projection onto the statistical model that can capture the distribution of the surface normal directions. However, the analysis of the distribution of surface

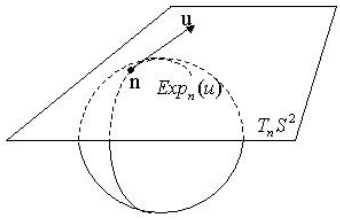


Figure 1: The exponential map.

normals cannot be effected in a linear way, because a linear combination of unit vectors (normals) is not itself a unit vector. Therefore, to construct the statistical model, we make use of principal geodesic analysis (PGA). PGA is a generalization of principle component analysis (PCA). For data residing on a Riemannian manifold, PGA is better suited to the analysis of directional data than PCA.

The idea of our work is to construct a statistical model from a set of ground-truth facial needle-maps, and combine the model and SFS to recover the training and testing needle-maps from the intensity images. Then by performing linear discriminant analysis (LDA) on the PGA parameters of the training needle-maps, we obtain the female and male models which are used to discriminate the genders of the testing faces. Because the SFS method satisfies data-closeness constraint, the recovered facial needle-maps not only represent the 3D shape, but also integrate the 2D texture information. Experiments show that this combination obtains better classification rate than using shape or texture alone.

The outline of the paper is as follows. Section 2 gives the idea of incorporating PGA into SFS to recover facial shapes. Section 3 reviews the LDA method and the probability based classification strategy. In section 4, firstly, a description of the data set and normalization is given, then, the experiment results and discussion are presented.

2 Facial Shape Recovery

We aim to recover the facial needle-maps using an iterative SFS which is augmented by a statistical model that captures variations of the surface normals. The surface normal $n \in R^3$ may be considered as a point lying on a spherical manifold $n \in S^2$, therefore, we turn to the intrinsic mean and PGA proposed by Fletcher et al. [8] to construct the statistical model.

2.1 Principal Geodesic Analysis

If $u \in T_nS^2$ is a vector on the tangent plane to S^2 at n and $u \neq 0$, the exponential map, denoted Exp_n, of u is the point, denoted $Exp_n(u)$, on S^2 along the geodesic in the direction of u at distance $\| u \|$ from n. This is illustrated in Fig. 1. The log map, denoted Log_n is the inverse of the exponential map.

The intrinsic mean is defined as $\mu = \arg\min_{n\in S^2} \sum_{i=1}^{N} d(n,n_i)$, where $d(n,n_i) = \arccos(n \cdot n_i)$ is the arc length. For a spherical manifold, the intrinsic mean can be found using the

gradient descent method of Pennec [11]. Accordingly, the current estimate $\mu^{(t)}$ is updated as follows: $\mu^{(t+1)} = \mathrm{Exp}_{\mu^{(t)}}(\frac{1}{N}\sum_{i=1}^{N}\mathrm{Log}_{\mu^{(t)}}(n_i))$.

In PGA each principle axis is a geodesic curve. In the spherical case this corresponds to a great circle. For a geodesic G passing through the intrinsic mean μ, π_G may be approximated linearly in the tangent plane $T_\mu S^2$: $\mathrm{Log}_\mu(\pi_G(n_1)) \approx \sum_{i=1}^{K} V^i \cdot \mathrm{Log}_\mu(n_1)$, where $V_1, \ldots V_K$ is an orthonormal basis for $T_\mu S^2$, which are the principal eigenvectors of the covariance matrix of the long vectors $U = [u^1|\ldots|u^K]$. $u^k = [u_1^k, \ldots, u_N^k]^T$ is the log mapped long vector of the k^{th} sample data, in our experiment, the k^{th} training needle-map.

We use the numerically efficient snap-shot method of Sirovich [13] to compute the eigenvectors of L. Accordingly, we construct the matrix $\hat{L} = \frac{1}{K}U^T U$, and find the eigenvalues and eigenvectors. The i^{th} eigenvector e_i of L can be computed from the i^{th} eigenvector $\hat{e}_i$of $\hat{L}$ using $e_i = U\hat{e}_i$. The i^{th} eigenvalue λ_i of L equals the i^{th} eigenvalue $\hat{\lambda}_i$ of $\hat{L}$ when $i \leq K$. When $i > K$, $\lambda_i = 0$. In our experiments, we use the $K-1$ leading eigenvectors of L as the columns of the eigenvector matrix (projection matrix) $\Phi = (e_1|e_2|\ldots|e_{K-1})$, where K is the number of training data.

Given a long vector $u = [u_1, \ldots, u_N]^T$, we can get the corresponding PGA parameters $b = \Phi^T u$. Given the PGA parameters $b = [b_1, \ldots b_S]^T$, we can generate a needle-map using: $n_p = \mathrm{Exp}_{\mu_p}((Pb)_p)$.

2.2 Incorporating PGA into SFS

The iterative SFS method used in our experiments satisfies a strict global constraint (projection onto the statistical model) as well as a hard local constraint (satisfaction of the image irradiance equation) in each iteration.

Let $I \in R^N$ denote the intensity image, according to Worthington and Hancock [17], when the surface reflectance follows Lambert's law, then the surface normal is constrained to fall on a cone whose axis is in the light source direction s and whose opening angle is $\alpha = \arccos I$. Therefore, the image irradiance equation $I = n \cdot s$ is satisfied by constraining the recovered surface normal to lie on the reflectance cone. While, the recovered surface normal is generated from the best fit PGA parameters, thus the constraint imposed by statistical model is as well satisfied.

So, there are two steps in each iteration (t). First the surface normal at pixel p is generated by: $n_p^{(t)} = Exp_\mu(u_p^{(t)})$, where $u_p^{(t)} = (\Phi b^{(t)})_p$, and $b^{(t)}$ is the current best fit PGA parameters. Then, to satisfy the data-closeness constraint, each surface normal is rotated back to its closest on-cone position by: $n_p^{(t+1)} = Exp_s(\arccos(I_p)\frac{Log_s(n_p^{(t)})}{\|Log_s(n_p^{(t)})\|})$. This is demonstrated in Fig. 2 which shows the tangent plane $T_s S^2$. This principal geodesic SFS algorithm can be summarized as follows:

1. Initialize: $N^{(0)} = \mu$, where μ is the intrinsic mean of the model. Set iteration $t = 0$.
2. Estimate PGA parameters: $b^{(t)} = \Phi^T \log_\mu(N^{(t)})$.
3. Update normals: $N^{(t+1)'} = Exp_\mu(\Phi b^{(t)})$, $N^{(t+1)''} = Exp_s(\arccos(I) .* \frac{Log_s(N^{(t)})}{\|Log_s(N^{(t)})\|})$, where the operator $.*$ denotes the component-wise product of any two vectors of the same length. Set $t = t + 1$.
4. Stop if $t >$max iterations. Set $N' = N^{(t)'}$, $N'' = N^{(t)''}$.

Upon convergence, the output N' is an instance of the statistical model. N'' is the closest on-cone position of N', therefore it satisfies data-closeness constraint. However,

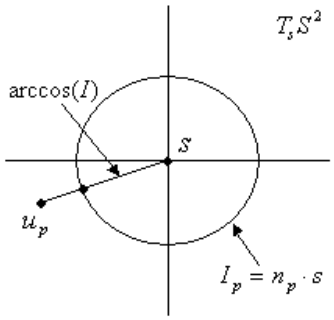


Figure 2: Restoring a normal to the closest position on the reflectance cone.

as stated in [14], the angular difference between N' and N'' is almost solely due to the variation in albedo at the eyes, eye-brows and lips. Therefore, N' can be considered more close to the shape of the face, while N'' combines the facial shape and texture information. Our experiments show the difference between the classification rates of using N' and N''. The albedo at each pixel can be estimated as well [14]: $\rho_p = \frac{I_p}{s \cdot N'_p}$.

3 Feature Extraction and Classification

The recovered facial needle-maps are represented by the PGA parameters: $b = \Phi^T \log_\mu(N)$. However, the parameter vectors inevitably contain information which is either redundant or irrelevant to the gender classification task. Therefore, we extract the most discriminant features for gender using certain discriminating analysis method on the PGA parameters, then the classification is performed on the extracted feature vectors. Linear Discriminant Analysis (LDA) is a simple and widely used discriminating analysis method, and it is especially suitable to the analysis in the PCA subspace. Therefore, in our gender classification task, we choose LDA as the discriminating analysis method.

3.1 Linear Discriminant Analysis

Linear Discriminant Analysis [1] utilizes the class information of the training data to find the vectors in the underlying space that best discriminate among classes. The transform matrix W is chosen in such a way that the ratio of the between class scatter and the within class scatter is maximized. Let the between class scatter be defined as: $S_B = \sum_{i=1}^{c} N_i(\mu_i - \mu)(\mu_i - \mu)^T$, and the within class scatter be defined as: $S_W = \sum_{i=1}^{c} \sum_{x_k \in X_i} (x_k - \mu_i)(x_k - \mu_i)^T$, where μ is the mean of all the samples, μ_i is the mean of class X_i, and N_i is the number of samples in class X_i. Then $W_{opt} = \arg\max_W \frac{|W^T S_B W|}{|W^T S_W W|} = [w_1, w_2, \ldots, w_m]$, where $\{w_i | i = 1, 2, \ldots, m\}$ is the eigenvectors of $S_W^{-1} S_B$ corresponding to the m largest eigenvalues.

Gender classification is a two-class problem, therefore the class number $c = 2$. The rank of S_w is at most $N - c$, where N is the number of training data. Therefore to avoid

S_w to be singular, the input PGA parameters are truncated to be $N-c$ dimensional by keeping the first $N-c$ components. Because there are at most $c-1$ nonzero generalized eigenvalues for $S_W^{-1}S_B$, the extracted feature vectors contain only one component in our experiments.

3.2 Classification

After the feature vectors of the training and testing faces have been extracted, we use the a posteriori class probabilities to classify testing faces to one of the genders.

Let $C = \{C_i | i = female, male\}$ denotes the two gender classes, x denotes the feature vector of any testing face. Then according to the Bayes law, the probability that x is of class C_i is:

$$P(C_i|x) = \frac{P(x|C_i)P(C_i)}{\sum_{i=female,male} P(x|C_i)} \tag{1}$$

We assume that the distribution of gender is Gaussian, and the mean and covariance of class C_i is μ_i, σ_i, the a priori probability is $P(C_i) = 0.5$. Then,

$$P(x|C_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp(-\frac{(x-\mu_i)^2}{2\sigma_i^2}). \tag{2}$$

The a posteriori probabilities $P(C_{female}|x)$ and $P(C_{male}|x)$ are computed by applying (2) and the a priori probabilities into (1). If $P(C_{female}|x) > P(C_{male}|x)$, then the face is classified as female. Otherwise, the face is classified as male.

4 Experiments and Discussion

In this section, we first introduce the data set and the normalization method used in our experiments. Then, we do experiments on the normalized data to evaluate the performance of gender classification based on the recovered facial needle-maps. In experiments, we first examine the performance of SFS. Then the results of gender classification based on the recovered needle-maps are given in comparison with the results based on intensity images. Finally, we compare our method with human classification. The results show that our gender classification strategy is competive to human classification and achieves better classification rates than using facial shape and texture information alone.

4.1 Data set and normalization

The data set used in our experiments contains 260 3D and corresponding 2D frontal face images (103 females and 157 males) selected from the University of Notre Dame Biometrics Database [9], [7]. To construct the statistical modal and apply the SFS method, the 3D and 2D face images are required to be normalized.

Geometric normalization is needed for the 3D scans which are a set of points $S = \{(x,y,z)\}$. Seven points are manually selected (as shown in Fig. 3): the inside and the outside corners of the left eye (1,2), the inside and the outside corners of the right eye (3,4), the nose tip (5), the middle of the lips (6), the chin point (7). The centers of the

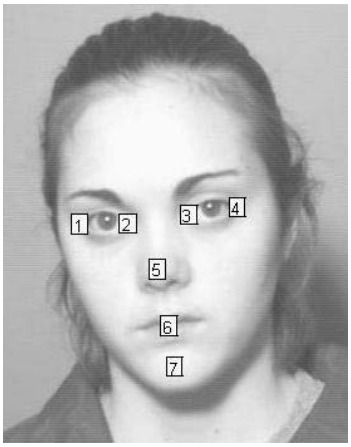


Figure 3: Seven control points.

left and the right eyes, denoted as El and Er, are calculated as the midpoints of 1 and 2, 3 and 4. We first rotate and translate the face scans so that the plane passing El, Er and 7 is perpendicular to the Z-axis, the line passing 1 and 2 is horizontal, and the XY position of 5 is (0,0). After that, we calculate five mean positions (El, Er, 5, 6, 7) for all the 260 scans, and use them as the reference points. Then, we scale the scans to make the distance between 1, 2 identical to the reference. The nose tip 5 gives the centerline for cropping a 100×114 region from the raw 3D scan to create a range image from the depth values. Then, we use the principal warps proposed in [2] to warp the range image so that the XY positions of the five points (El, Er, 5, 6, 7) are identical to those of the reference points. Then linear interpolation is used to fill the holes.

The geometric normalization for the 2D images is almost the same as for the 3D scans, except that the rotation and cropping are performed in XY plane only. Besides the geometric normalization, the 2D images need brightness normalization as well. First, the colored images are converted into greyscale, the intensity at each pixel is the mean value of the three color channels. Then, the intensity contrast is stretched to normalize for ambient lighting variations. Finally, photometric correction and specularity subtraction are applied using the method proposed in [12] in order to improve the results of SFS.

After the normalization, the facial needle-maps are calculated from the 3D face scans. Because the cropping step involves re-sampling, some noises are introduced into the needle-maps. Examples of the rendered needle-maps and the normalized 2D images are shown in Fig. 4.

4.2 Experiment Results

In the experiments, 200 normalized face images (needle-maps and corresponding intensity images) are randomly chosen for training, while the remaining 60 are used for testing. In this way, we run the system 6 times and obtain 6 different training and testing data sets. We first apply PGA to the training facial needle-maps to construct the statistical model

Figure 4: Examples of normalized 2D images (1st row) and surface normals (2nd row).

for SFS. Then, the principal geodesic SFS is used on the training intensity images to get the recovered facial needle-maps for the training data. The recovered needle-maps for the testing data are obtained in the same way. We represent the recovered needle-maps in PGA parameters and apply LDA to extract the feature vectors. The Gaussian models for female and male are constructed from the training feature vectors, and are used to discriminate the gender for the testing faces.

Performance of SFS The recovered needle-maps (N' and N'') and surfaces, together with the estimated albedos of two face images (one male and one female) are shown in Fig. 5. The ground-truth needle-maps and surfaces are also given for comparison. Since N'' satisfy data-closeness, it would appear identical to the image when rendered with a frontal light source. For this reason, we show N'' reilluminated with a light source moved 30° from the viewing directionalong along the positive x-axis. For comparison convenience, N' and the ground-truth needle-map are also reilluminated in the same way. From the figure, we can see the recovered needle-maps and the surfaces are similar to the ground-truth ones, especially at the cheek and mouth regions in which some gender information is encoded. Both the albedos and N' convey some gender information. However, more gender information is conveyed in N'' which implicitly combines the facial shape and texture.

There are noises in the recovered needle-maps. The reason is that the ground-truth data are noised because of the re-sampling in the alignment, and the statistical model used in SFS is constructed from the noised ground-truth data. Therefore the model captures the noises and introduces it to the recovered needle-maps. Modification of the re-sampling in data alignment to reduce the noise will be needed in our future work.

Gender Classification Results The gender classification rates are shown in Table 1. The experiment results show that both 2D facial texture (intensity images) and 3D facial surface shape (N') shows some capability for gender classification. However, using the recovered facial needle-maps satisfying data-closeness - the implicit combination of

Figure 5: Examples of the results of SFS. From left to right, the columns are: the input intensity images, the estimated albedos, the recovered N', the recovered N'', the ground-truth needle-maps. The 2nd and 4th rows are the recovered surfaces of the according needle-maps.

facial texture and shape - achieves better classification rates than using texture or shape individually. Because of the recovery of shape from images, the classification using N' and N'' are more computationaly expensive than using the intensity images. However in our experiments the recovery of the shapes for 200 images takes less than 5 minutes.

	Female	Male	Overall
Intensity	$83.2\% \pm 0.161$	$89.8\% \pm 0.069$	$87.2\% \pm 0.092$
N'	$68.2\% \pm 0.079$	$70.6\% \pm 0.048$	$69.7\% \pm 0.044$
N''	$95.0\% \pm 0.062$	$92.7\% \pm 0.046$	$93.6\% \pm 0.040$

Table 1: Gender classification performance.

We use one randomly selected training and testing data set to compare the results of our method using N'' with the results of human classification. We present the 60 testing intensity images to 10 subjects (4 males and 6 females) who are not familiar with the images and are inocent of our research. Their average correct classification rates are calculated. The results are shown in Table 2. From the table, we can see that using the same intensity images (cropped and hair removed), the classification rates achieved using our method are similar to (even better than) the results of human classification.

	Female	Male	Overall
Machine	89.3%	96.9%	93.3%
Human	86.1%	95.6%	91.2%

Table 2: Comparison with human classification.

5 Conclusion

In this paper, we present a gender classification method based on the 2.5D facial needle-maps recovered using SFS. We incorporate PGA into SFS to guarantee the projection onto a statistical model as well as satisfy the data-closeness constraint. The recovered needle-maps implicitly combine the facial shape and texture information. Experiment results show that using the recovered facial needle-maps satisfying data-closeness achieves better gender classification rate than using facial texture or shape information individually.

However, there are some problems that require further investigation. First, modifications of the cropping method in data alignment are needed to reduce the noises caused by re-sampling. Second, the SFS method used is only applicable to grey scale images. The extension to recover needle-maps from color images may improve the gender classification further. Third, apply LDA to the training data is a supervised learning. In the future, we will apply unsupervised learning method to construct models for different genders.

References

[1] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. on PAMI*, 19(7), 1997.

[2] F.L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. on PAMI*, 11(6), 1989.

[3] V. Bruce, A.M. Burton, E. Hanna, P. Healey, O. Mason, A. Coombes, R. Fright, and A. Linney. Sex discrimination: how do we tell the difference between male and female faces? *Perception*, 22:131–152, 1993.

[4] S. Buchala, N. Davey, R.J. Frank, and T.M. Gale. Dimensionality reduction of face images for gender classification. *Department of Computer Science, The University of Hertfordshire, UK, Technical Report 408*, 2004.

[5] S. Buchala, N. Davey, T.M. Gale, and R.J. Frank. Principal component analysis of gender, ethnicity, age, and identity of face images. In *IEEE ICMI 2005*, 2005.

[6] A.M. Burton, V. Bruce, and N. Dench. What's the difference between men and women? evidence from facial measurement. *Perception*, 22:153–176, 1993.

[7] K. Chang, K.W. Bowyer, and P.J. Flynn. Face recognition using 2d and 3d facial data. *ACM Workshop on Multimodal User Authentication*, pages 25–32, 2003.

[8] P.T. Fletcher, S. Joshi, C. Lu, and S.M. Pizer. Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE Trans. on Medical Imaging*, 23:995–1005, 2004.

[9] P.J. Flynn, K.W. Bowyer, and P.J. Phillips. Assessment of time dependency in face recognition: An initial study. *Audio and Video-Based Biometric Person Authentication*, pages 44–51, 2003.

[10] X. Lu, H. Chen, and A.K. Jain. Multimodal facial gender and ethnicity identification. In *ICB 2006*, pages 554–561, 2006.

[11] X. Pennec. Probabilities and statistics on riemannian manifolds: A geometric approach. *Technical Report RR-5093, INRIA*, 2004.

[12] A. Robles-Kelly and E.R. Hancock. Estimating the surface radiance function from single images. *Graphical Models*, 67:518–548, 2005.

[13] L. Sirovich. Turbulence and the dynamics of coherent structures. *Quart. Applied Mathematics*, XLV(3):561–590, 1987.

[14] W.A.P. Smith and E.R. Hancock. Recovering facial shape and albedo using a statistical model of surface normal direction. *10th IEEE ICCV*, 1:588–595, 2005.

[15] T. Wilhelm, H.J. Bohme, and H.M. Gross. Classification of face images for gender, age, facial expression, and identity. In *ICANN 2005*, pages 569–574, 2005.

[16] T. Wilhelm, H.J. Bohme, H.M. Gross, and A. Backhaus. Statistical and neural methods for vision-based analysis of facial expressions and gender. In *IEEE ICSMC 2004*, pages 2203–2208, 2004.

[17] P.L. Worthington and E.R. Hancock. New constraints on data-closeness and needle map consistency for shape-from-shading. *IEEE Trans. on PAMI*, 21(12):1250–1267, 1999.

Who are you? – real-time person identification

Nicholas Apostoloff and Andrew Zisserman
Department of Engineering Science, University of Oxford
{nema,az}@robots.ox.ac.uk

Abstract

This paper presents a system for person identification that uses concise statistical models of facial features in a real-time realisation of the cast identification system of Everingham *et al*. [7]. Our system integrates the cascaded face detector of Viola and Jones with a kernel-based regressor for face tracking, which is trained on-line when new people are detected in the video stream. A pictorial model is used to compute the locations of facial features, which form a descriptor of the person's face. When sufficient samples are collected, identification is performed using a random-ferns classifier by marginalising over the facial features. This confers robustness to localisation errors and occlusions, while enabling a real-time search of the database. These four different processes communicate within a real-time framework capable of tracking and identifying up to 5 people in real-time on a standard dual-core 1.86GHz machine.

1 Introduction

Have you ever been greeted by a person that you have only met briefly, perhaps at a previous BMVC, and wished that you had a magic little device that whispered their name in your ear? The goal of this paper is to implement such a device using a modern multi-core computing architecture, with the only input a video stream obtained from a standard web-cam. To make the problem even more challenging, this must be accomplished in real-time and be able to handle variations in their appearance due to lighting, scale, pose, expressions, image quality, motion blur, etc. The practical goal is the identification of people as they approach the camera, which has the fortunate consequence that we need only consider near-frontal faces. This has important applications in humanoid robotics, surveillance, human-computer interaction, tourism or as an aid for those with impaired vision (or memory).

We build this device using standard off-the-shelf components. The input video stream is captured by a web-cam which may hang around a person's neck or be embedded in their clothing. A standard dual-core laptop, which may be carried in a backpack, is used to process the video stream. The system, which we name *IDU*, stores a repository of people that you have met previously, including their name, the date and location that you met them and a video of that initial meeting. This information is collected automatically by the system when the user chooses to add a new person to the repository. A short 250 frame video is captured and then processed to extract an appearance model of the person.

1.1 Outline

The task of person identification can be broken into three sequential stages: detection (§3.1), tracking (§3.2) and identification (§3.3)—see figure 1. The goal is then to design

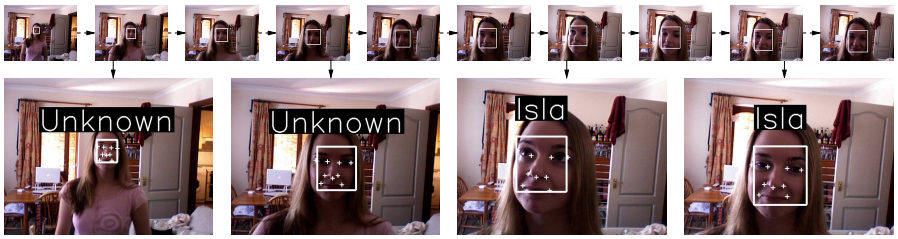


Figure 1: **Overview and example of the real-time IDU system**. The detection, tracking and identification of each person in a web-cam video stream is accomplished in three threads. When a new person is detected by the detection thread, a tracking thread is instantiated to track that person. (top) The tracking thread estimates the pose of the person's face at 15fps and proceeds sequentially (indicated by the dashed lines). (bottom) The feature localization and identification thread depends only on instances of a person's face given by the tracker (indicated by the solid arrows) and runs at 2fps. As 10 facial descriptors are required for reliable recognition, identification takes up to 5s per person. The crosses indicate the facial features used for recognition.

a system that can accurately track multiple people at 15fps and identify them in a timely manner. In our case, it takes approximately 5s to identify a person from the moment they are first detected. An additional design constraint is to ensure that the database search and the entry of new people into the database is scalable.

We build on previous approaches that have used frontal face detections in combination with textual information to automatically identify cast members in TV or film material, in an off-line, batch process [7]. In particular we make three contributions. First, we have implemented a real-time realisation of [7] for person recognition. This in itself, required a significant modification of the original system to guarantee real-time performance. Second, the tracking method used in [7] to group face detections in order to label cast members, does not lend itself to a real-time implementation. To mitigate this computational complexity, a simple kernel based regressor is developed that can be trained for each detected person and is capable of tracking the pose over time. Training the tracker for each individual confers the benefit of greater accuracy while requiring fewer computational resources. Third, a random-ferns classifier is used to identify individuals by marginalizing over a set of facial features, alleviating the effect of localisation errors and occlusions. Random-ferns [15] are a variation on the random-forest classifier first introduced in [1] and developed further in [5]. Their recent popularity owes much to the tracking application of [11]; however, they have also been applied to object recognition in [14, 25, 26]. The advantage of randomized trees/ferns is that they are much faster in training and testing than traditional classifiers (such as an SVM), without a loss in performance.

We show that our real-time classification method outperforms the exhaustive search criteria of the original paper, while requiring far fewer computation resources. A typical search using the random-ferns classifier takes approximately 0.3ms per face in a track, while the exhaustive *min-min* search of [7] takes approximately 200ms per face[1].

[1]The timing results depend on the number of people in the database and in the case of the *min-min* search, the number of instances of each person in the database. In this case, the timing results are presented for the Buffy the Vampire Slayer dataset described in §5 when the system is trained using episode 05-02.

2 Previous work

Automatic person recognition systems have been approached in two distinct strands of research in the literature. A number of authors attempt to identify people in *real-time* systems for surveillance [6] and human-computer interaction [21] using facial appearance [16], gait [17], thermal imagery [2] and even clothing [20], but frequently constrain the environment in which they operate (*i.e*. stationary cameras [17], fully frontal faces and consistent lighting [16, 22], etc.). In a separate strand of research, *off-line batch* processing systems have been used to identify cast members in TV episodes or movies by clustering frontal faces [10], or to retrieve instances of particular characters [3, 19] in a similar fashion to internet search engines. These methods have generally shown good performance over large volumes of difficult data with dramatically varying conditions and we have chosen to adapt these into a real-time application.

Review of Everingham *et al*. [7]. The recent cast identification system of Everingham *et al*. [7] showed excellent performance over the difficult conditions often found in TV or film material and is the basis upon which this real-time system is built. While they consider the more difficult problem of automatically building a database of cast members using both visual and textual information, we review only their contributions relevant to the work presented here.

In [7], an off-line face tracking approach first computes all face detections in the video stream using a cascaded face detector [23] with a high threshold designed to reduce the number of false positives. To reduce the computational burden of identifying each individually detected face[2], face detections linked by features points tracked using the Kanade-Lucas-Tomasi [18] tracker are grouped together. In addition to reducing the volume of data to be processed, this also allows stronger appearance models to be built for each character, as multiple instances of the character are grouped in each track. For this reason, the granularity of the identification procedure is reduced from matching individual faces to matching sets of faces grouped by tracks over the video sequence.

The appearance models are built from 13 local exemplar-based descriptors at the corners and centres of the eyes, nose and mouth. Using local descriptors (in this case, normalized image patches) confers the benefit of robustness to pose variation, lighting and partial occlusion when compared with global descriptors such as eigenfaces [16]. A descriptor for each face $\mathbf{F}$ is then the concatenation of each of the facial feature descriptors $\mathbf{f}_i$. The features are located using a generative model of the feature positions combined with a discriminative model of feature appearance in an extension of the pictorial tree in [9]. Given a database of face tracks, each consisting of a set of facial descriptors $\mathscr{F}_t = \{\mathbf{F}_i\}_t$ with a corresponding label λ_t, a new face track $\mathscr{F}$ is matched against the set of all faces $\mathscr{F}(\lambda_i)$ with the label λ_i using a *min-min* metric

$$d(\mathscr{F}, \lambda_i) = \min_{\mathbf{F}_j \in \mathscr{F}} \min_{\mathbf{F}_k \in \mathscr{F}(\lambda_i)} ||\mathbf{F}_j - \mathbf{F}_k||. \quad (1)$$

For simplicity, we ignore the clothing descriptor they use in combination with the facial descriptors. For a given track $\mathscr{F}$, the likelihood that it corresponds to face λ_i is

[2] A typical episode of a TV series contains over 20,000 face detections, even though these generally arise from a few hundred "tracks" of a particular character.

then $p(\mathscr{F}|\lambda_i) = Z^{-1}\exp\{-d(\mathscr{F},\lambda_i)^2/(2\sigma^2)\}$. The posterior is then obtained by applying Bayes rule to get $p(\lambda_i|\mathscr{F}) = p(\mathscr{F}|\lambda_i)/\sum_j p(\mathscr{F}|\lambda_j)$. A face track is then labelled with the λ_i that maximizes the posterior.

A brief introduction to random forests/ferns. A random forest multi-way classifier consists of a number of trees, with each tree grown using some form of randomization. The leaf nodes of each tree are labelled by estimates of the posterior distribution over the image classes. Each internal node contains a test that splits the space of data to be classified. An image is classified by sending it down every tree and aggregating the reached leaf distributions. Randomness can be injected at two points during training: in sub-sampling the training data so that each tree is grown using a different subset; and in selecting the node tests. In this paper we do only the later.

Suppose that T is the set of all trees, C is the set of all classes and L is the set of all leaves for a given tree. During the training stage the posterior probabilities ($P_{t,l}(\lambda(\mathbf{F}) = c)$) for each class $c \in C$ at each leaf node $l \in L$, are found for each tree $t \in T$. These probabilities are calculated as the ratio of the number of descriptors $\mathbf{F}$ of class c that reach l to the total number of descriptors that reach l. To reduce the effect of unbalanced datasets, the contribution of each descriptor of class c is further normalized by the number of samples of class c in the training dataset. In order to classify a new test descriptor, it is passed down each random tree until it reaches a leaf node. All the posterior probabilities are then averaged over the trees and the arg max is taken as the classification of the input descriptor.

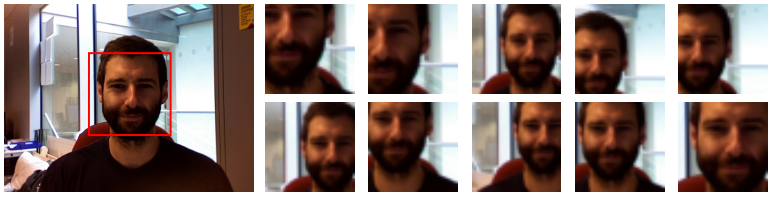
To increase the speed of the random forest Ozuysal *et al*. proposed random-*ferns* classifiers [15]. In the case of ferns, there are an ordered set of nodes, and the node test is applied to the whole training data set. In contrast, in random forests only the data that falls in a child node is taken into account in the test. As in random forests, leaves store the posterior probabilities. At each node in the fern set, a test gives a binary result. The result of each test and the ordering on the set defines a binary code for accessing the leaf node. So if a fern has N nodes, it will have 2^N leaves. The advantage of ferns over forests is that it is not necessary to store the intermediate nodes (of a tree).

3 Visual processing methods

This section describes the three stages of visually identifying a person: detection, tracking and recognition.

3.1 Face detection

The first stage in processing is face detection where a frontal face detector is run once per second to obtain new tracks. This detector is based on the Viola-Jones [23] cascaded frontal face detector and enhancements by Lienhart [12] and is moderately fast, being able to process each frame in ≈ 200 ms. The output of the cascaded face detector is a set of face poses $\mathbf{p}_d = (x_d, y_d, w_d)^\top \in \mathscr{P}_D$ where the pose is a vector containing the spatial coordinates (x_d, y_d) of the centre of the face and its scale w_d. When a new face $\mathbf{p}_d$ is detected, it is compared to all faces currently being tracked $\mathbf{p}_t \in \mathscr{P}_t$ and a new track is only started if $\mathbf{p}_d$ does not overlap any $\mathbf{p}_t$ by more than a quarter of its area.



(a) Face detection (b) Training samples

Figure 2: **Training the regressor from face detections**. Given a set of input faces from the localized face detector, (a), a set of 192 $(\mathbf{v},\tilde{\mathbf{t}})$ pairs (b) are artificially sampled over the state-space of face poses for training.

3.2 Face tracking

Inspired by the relevance vector machine (RVM) tracker of Williams *et al*. [24] we have implemented a regularized kernel-based face tracker using radial basis functions (RBF). This choice, over that of the RVM, is motivated by the lack of sparsity that is generally present within an RVM trained for face tracking. This is supported by the general regression tracker of Mayol and Murry [13] who also claim that the sparsity induced by the RVM is detrimental to tracking performance because the set of kernels after training is incomplete.

After a new face $\mathbf{p}_d$ has been detected, the tracker enters a training state where it collects samples of the person's face over a number of frames using a localized version of the cascaded face detector, where the search region at frame $i+1$ is centred on the detected face from frame i and has a width equal to $1.2w^{(i)}$. After a preset number of face exemplars are detected, a set of training image/target pairs $\{\mathbf{v},\tilde{\mathbf{t}}\}$ is generated by sampling the 3D space of possible translations and scales over a uniform grid with $N = 192$ elements (see figure 2). Here, $\tilde{\mathbf{t}} = (dx, dy, dw)$ and lies in the range $(\pm 10, \pm 10, \pm 5)$ in a canonical reference frame where the face image has dimensions 40×40 pixels. Each training image patch $\mathbf{v}$ is then scaled to the reference frame, Gaussian blurred and histogram normalized to reduce the effect of lighting. A kernel-based regressor is then trained for each dimension of the state-space separately.

The kernel-based regressor models the target variable $t(\mathbf{v},\mathbf{w})$ as a linear combination of a fixed number of $N+1$ nonlinear functions $\phi_j(\mathbf{v})$ of the input variables $\mathbf{v}$

$$t(\mathbf{v},\mathbf{w}) = \sum_{j=0}^{N} w_j \phi_j(\mathbf{v}) = \mathbf{w}^\top \phi(\mathbf{v}), \tag{2}$$

where $\mathbf{w} = (w_0,\ldots,w_N)^\top$ and $\phi = (\phi_0,\ldots,\phi_N)^\top$ with $\phi_0 = 1$ to account for bias. Here, N is the number of training samples and the kernel function ϕ_j is a radial basis function centred on the training vector $\mathbf{v}_j$. Given a set of N $(t_j,\mathbf{v}_j)$ training pairs, the weights $\mathbf{w}$ can be estimated by placing a quadratic regularizer on the weights and by minimizing the sum-of-squares error function as in *ridge regression* [4] to get $\mathbf{w}_R = (\lambda\mathbf{I} + \Phi^\top\Phi)^{-1}\Phi^\top\mathbf{t}$. Here, the ith and jth element of Φ is the result of applying the jth nonlinear function to the ith input vector $\phi_j(\mathbf{v}_i)$ and $\mathbf{t} = (t_1,\ldots t_N)$, and λ is a system parameter that controls the smoothness of the solution.

The input to each regressor at frame $i+1$ is an image patch extracted from the current image *at the location of the face from frame i*, $\mathbf{p}_i = (x_i, y_i, w_i)$. This is extracted in the same manner used to calculate the training vectors. The output of the three regressors at frame

$i+1$ is then an estimate of the change in the pose $(dx, dy, dw)^\top$ in the canonical reference frame. Hence, the pose at frame $i+1$ is then $\mathbf{p}_{i+1} = (x_i + \hat{dx}, y_i + \hat{dy}, w_i + \hat{dw})$, where $(\hat{dx}, \hat{dy}, \hat{dw})$ is the transformation of (dx, dy, dw) from the canonical reference frame into image space.

The regression framework estimates scale and translational offsets, but does not estimate rotations of the head; however, depending on the variability of the training vectors, it can accurately track a person's face with slight out-of-plane rotations (figure 1). This exhibits similar fail-case scenarios to the RVM tracker where tracks drift off the target when the person's head is rotated more than 30^{o} about the vertical axis. Rotations about the horizontal axis tend to be small and do not effect the results significantly.

3.3 Face recognition

The face recognition system uses the pictorial structure model of Everingham *et al*. [7] to locate 9 facial features at the corners of the eyes, nose and mouth (figure 1). Four additional features are added at the centre of both eyes, mouth and nose. The face region defined by the facial features is normalized with respect to a canonical face to reduce the effects of scale and out-of-plane rotations of the head. An affine transformation is computed between the canonical feature set and the facial features. Using the affine transformation, regions that were circular in the canonical reference frame are extracted from the corresponding elliptical regions in the tracked face. These 13 image patches ($\mathbf{f}_{1\ldots13}$) are then normalized to have zero mean and unit variance, and are concatenated to form a single vector that represents the persons face ($\mathbf{F}$) as in §2. The canonical reference frame has dimensions 80×80 pixels while the extracted image patches have a diameter of 15 pixels.

To classify the faces we use a random-ferns classifier with 40 ferns of 17 levels, where the test at node n is a simple comparison between two elements (p_n and q_n) of the descriptor $\mathbf{F}$, which are chosen at random when building the tree. In this case, elements $\mathbf{F}(p_n)$ and $\mathbf{F}(q_n)$ are compared using the less-than operator ($\mathbf{F}(p_n) < \mathbf{F}(q_n)$).

As mentioned previously, the result of sending a new test descriptor down a random-ferns classifier is a posterior distribution over the classes ($P(\lambda(\mathbf{F}) = c)$). In the simplest case, the label given the test descriptor can be taken as the argmax_c of this distribution; however, in the situation here, we have the additional knowledge of multiple faces in each track $\mathbf{F}_i \in \mathscr{F}$. Hence, two options are available when classifying a track. First, we can take the max over the set of posteriors and faces

$$\lambda_{\text{max}-\text{max}} = \underset{c \in C}{\text{argmax}} \max_{i \in \mathscr{F}} P(\lambda(\mathbf{F}_i) = c), \tag{3}$$

which we label the *max-max* result. In the second method, we marginalize over the faces in the data set to get the *max-sum* result

$$\lambda_{\text{max}-\text{sum}} = \underset{c \in C}{\text{argmax}} \sum_{i \in \mathscr{F}} P(\lambda(\mathbf{F}_i) = c). \tag{4}$$

In an alternative approach, we train a random-ferns classifier for each facial feature. To classify a new descriptor $\mathbf{F}$, we first marginalize over the facial features $\mathbf{f}_i$ by summing the posteriors returned from each classifier $P(\lambda(\mathbf{f}_i) = c)$, and then apply either the *max-max* or *max-sum* variant. We label these classifiers *max-max-sum* and *max-sum-sum* respectively, and compare the classification performance of these variants in section 5.

4 Architecture

With the introduction of multi-core computing architectures, it becomes possible to process multiple CPU intensive tasks simultaneously, enabling a new branch of real-time applications for video processing. The IDU algorithm takes advantage of these architectures to capacitate real-time person tracking and identification, by segregating the time-sensitive tracking tasks from the slower identification task in a multi-threaded environment. While the kernel-based regressor presented in this paper can efficiently track multiple people in real-time on a single core, the localization of the facial features required for recognition is an expensive process, requiring up to 500ms per frame on a 1.86GHz machine. To mitigate this complexity, the crucial tracking tasks are computed on separate high-priority threads, while relegating the slow feature localization and identification tasks into the background on low-priority threads (figure 1). In live experiments, up to 5 people can be tracked at 15fps and while feature localization can only be computed at 2fps, only 10 faces are required for recognition, and reliable recognition can then be computed in less than 5s per person.

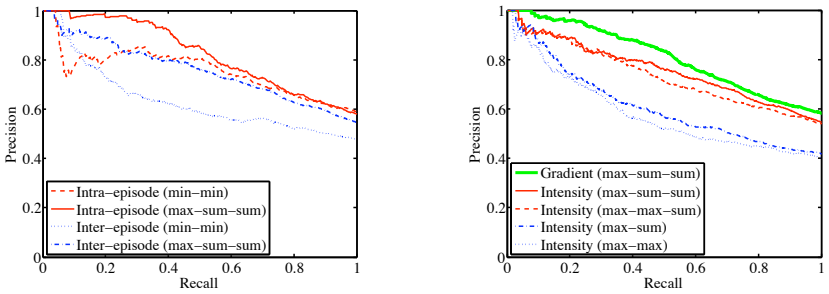
The detection, tracking and subsequent identification of a person proceeds in three phases. First, a global face detection thread using a variant of the Viola-Jones cascaded face detector runs once per second to detect new faces. Second, when a tracking thread has been initialized with a newly detected face, it enters a training state where it collects $N = 12$ face samples by performing a localized cascaded face detection. When enough face samples are collected, it trains a kernel-based regressor for each dimension of the state-space (*i.e*. the pose). Depending on the number of training vectors used, this can track an individual face at a cost of around 13ms per frame. Third, each tracked face is processed by an instance of the facial feature localisation algorithm in a separate thread and the result of this is a set of face descriptors $\mathscr{F}$. When a suitable number of descriptors is collected to make a model of a face (typically more than 10), the database is searched for a matching face model.

Lost tracks are handled directly by the identification system, which returns a score from the pictorial structure used to locate facial features. If subsequent faces over a period of 2 seconds in a track return low scores from the identification module, then the track is deemed lost and terminated.

While, this system has been designed to track and identify multiple people in real-time, it must also provide a satisfying user experience. To accomplish this it is necessary to separate the processing threads from the GUI and hence, there is a total of $2+2N$ threads running when tracking N people. This includes one thread for the GUI, one thread for the face detector and a separate tracking and identification thread for each person.

5 Results and discussion

In this section we compare various methods of training and scoring the random-ferns classifier, with the previously used *min-min* classifier of [7], and compare on the data they used. This consists of episodes 2 and 5 from season 5 of Buffy the Vampire Slayer, and contains 2462 tracks over 12 people. The tracks vary in length from 1 to 404 frames, and there are 53032 labelled face detections in the database. Everingham *et al*. provide their tracked ground truth data for these videos at [8]. In the case of processing the Buffy videos, the experimental setup was exactly the same as the IDU system except that the



(a) Comparing the *min-min* search with the random-ferns classifier

(b) Random-ferns classifier using **inter-episode** training

Figure 3: **Precision-recall curves for intra and inter-episode experiments**. Recall is the proportion of tracks assigned a label at a given confidence level, while precision is the proportion of those tracks assigned correct labels. (a) compares the *min-min* distance with the random-ferns classifier for *intra*-episode and *inter*-episode experiments. (b) compares the different configurations of the random-ferns classifier in *inter*-episode experiments and also compares gradient descriptors versus normalized intensity patches for the best configuration (*max-sum-sum*). See text for details.

video stream was read from disk instead of the web-cam.

Exemplar-based descriptors have performed well in experiments using TV or film material where all of the data (*i.e*. all face detections and tracks) is available for categorization [3, 7, 10, 19]. In these cases, many shot transitions are of the form ABAB, where A and B are two different shots and the scene alternates between the two. The remarkable performance of these techniques is partly the result of the repetitive nature of these shot transitions. The repeated shots contain face detections that are almost identical and can be matched accurately using the normalized patch descriptors and simple metrics; however, when trained and tested on separate episodes (or even different parts of the same episode) their performance degrades (see below). Consequently, these descriptors do not generalize well by themselves to our target application where a person's appearance can change significantly from that contained in the database. We mitigate these deficiencies by using a variation of the random-ferns classifier and by marginalizing over the facial features and faces in a track.

Two experimental scenarios are considered. First, we train the classifiers using 160 tracks from episode 05-02 which were obtained in [7] using speaker detection and test the classifier on all other tracks that are at least 10 frames long (*intra*-episode). This is the same experimental scenario examined in [7] except that no clothes descriptor is used and the speaker detected tracks are not included in the precision-recall curves. The speaker detection data contains 9.5% errors, and has 12 characters, one of which is an "other" class containing a mixture of characters. Second, we use all the tracks from a single episode to train the classifiers and then test them on a different episode (*inter*-episode). In figure 3(a) we compare the results of the *min-min* classifier of Everingham *et al*. with the *max-sum-sum* random-ferns classifier. The *min-min* classifier performs well in the intra-episode experiment due partly to the ABAB shots that are present; however, its performance degrades in the inter-episode experiments for larger recalls. In both cases, the *max-sum-sum* classifier outperforms the *min-min* classifier by a significant margin. For example, in the intra-episode experiments, the *max-sum-sum* obtained $97 \pm 2\%$ precision at 20% recall, while the *min-min* classify obtained 81% precision. In Figure 3(b), we compare the dif-

ferent variations of the random-ferns classifier and clearly see that marginalizing over the facial features again improves performance. This may be explained by the stability of the feature localisation algorithm and the effects of occlusion. It was found that while the eyes were quite stable features, those around the mouth and nose were quite unstable. Unstable features would naturally be evenly distributed over the leaf nodes when training the random-ferns classifier resulting in flat posteriors. On the other hand, stable features would frequently take the same paths down the random-ferns and would result in peaks in the posterior distributions. In a similar fashion, occluded features would traverse arbitrary paths in the classifier leading to leaf nodes that were untouched by the training data. The technique was further extended to use gradient descriptors instead of the normalized image patches, which further increased the precision. The gradient descriptors were constructed by concatenating dI/dx and dI/dy computed over each image patch.

Similar trends as found in [11, 15] were observed in the classification performance when increasing the number of ferns and the number of levels in each fern. The results improve significantly as the number of ferns approaches 30, but the improvement tapers off around 40, while a gradual improvement is observed when increasing the number of levels in the ferns above 15. This is limited however, by the prodigious memory requirements of storing the posteriors and 17 levels was found to be a good compromise.

Figure 4 shows a sample set of identification results using the *max-sum-sum* classifier. The first two rows show correct labelling of the data for a wide variety of scenarios with multiple people and varying lighting conditions and head pose. The bottom row shows a number of failure-scenarios which often result from the face detector not firing when there is a non-frontal face present or where there is extremely poor lighting. The final two images depict a scenario where Buffy was initially incorrectly labelled, but where the error was later rectified when additional face samples were collected.

It is emphasized again that while the results presented here were obtained by processing a video stream from disk, they were computed using the same real-time algorithm that is described in section 4 and shown in figure 1.

6 Conclusion

This paper has presented a system for person identification that is a real-time realization of [7]. Integrating a cascaded face detector with a kernel-based regressor that can track up to 5 people in real-time provides the basis for a person identification thread that runs in the background. We have shown that a random-ferns classifier achieves a far better classification performance than the *min-min* classifier of [7].

At this stage, visual feedback for testing is provided on the laptop display, but it is left for future work to implement auditory feedback via headphones. It is envisioned that these standard components could be combined into an ultra-portable device such as a PDA and the input/output embedded directly in the apparel of the person carrying the device. Later versions will automatically add a new person when they do not match anybody currently in the repository.

Acknowledgements. This work was supported by an EPSRC Platform grant.

References

[1] Y. Amit and D. Geman. Shape quantization and recognition with randomized trees. *Neural Computation*, 9:1545–1588, 1997.

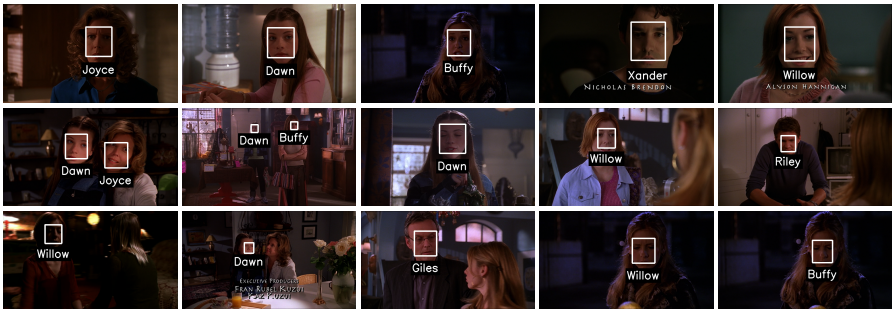


Figure 4: **Sample identifications from episode 05-05**. The top two rows show correctly matched faces, while the bottom row shows some sample failure cases. See text for details.

[2] O. Arandjelović, R. Hammoud, and R. Cipolla. On person authentication by fusing visual and thermal face biometrics. *In Proc. AVSS*, pages 50–56, November 2006.

[3] O. Arandjelovic and A. Zisserman. Automatic face recognition for film character retrieval in feature-length films. In *Proc. CVPR*, pages 860–867, 2005.

[4] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., USA, 2006.

[5] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[6] Y.-T. Chien, Y.-T. Huag, S.-W. Jeng, Y.-H. Tasi, and H.-X. Zhao. Real-time surveillance system by use of the face understanding technologies. In *Proc. DICTA*, 2003.

[7] M. Everingham, J. Sivic, and A. Zisserman. Hello! my name is... buffy – automatic naming of characters in tv video. In *Proc. BMVC.*, 2006.

[8] M. Everingham, J. Sivic, and A. Zisserman. Automatic naming of characters in tv video. `http://www.robots.ox.ac.uk/~vgg/research/nface/`, 2007.

[9] P. Felzenszwalb and Huttenlocher D. Pictorial structures for object recognition. *Intl. Journal of Computer Vision*, 61(1):55–79, January 2005.

[10] A. W. Fitzgibbon and A. Zisserman. On affine invariant clustering and automatic cast listing in movies. In *Proc. ECCV*, volume 3, pages 304–320. Springer-Verlag, 2002.

[11] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *IEEE PAMI*, 28(9):1465–1479, 2006.

[12] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *Proc. IEEE ICIP*, volume 1, pages 900–903, September 2002.

[13] W. W. Mayol and D. W. Murray. Tracking with general regression, 2007. In press. See `http://www.cs.bris.ac.uk/~wmayol/research/index.html`.

[14] F. Moosmann, B. Triggs, and F. Jurie. Fast discriminative visual codebooks using randomized clustering forests. In *NIPS*, 2006.

[15] M. Ozuysal, P. Fua, and V. Lepetit. Fast keypoint recognition in ten lines of code. In *Proc. CVPR*, 2007.

[16] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *Proc. CVPR*, pages 84–91, 1994.

[17] N. M. Rajpoot and Masood K. Human gait recognition with 3d wavelets and kernel subspace projections. In *Proc. HAREM*, Oxford (UK), September 2005.

[18] J. Shi and C. Tomasi. Good features to track. In *Proc. CVPR*, pages 593–600, 1994.

[19] J. Sivic, M. Everingham, and A. Zisserman. Person spotting: video shot retrieval for face sets. In *Proc. CIVR*, 2005.

[20] J. Sivic, C. L. Zitnick, and R. Szeliski. Finding people in repeated shots of the same scene. In *Proc. BMVC.*, 2006.

[21] K.-T. Song and W.-J. Chen. Face recognition and tracking for human-robot interaction. In *IEEE Intl. Conf. on Systems, Man and Cybernetics*, volume 3, pages 2877–2882, 2004.

[22] M. Turk and A. P. Pentland. Face recognition using eigenfaces. In *CVPR*, pages 586–591, 1991.

[23] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. CVPR*, pages 511–518, 2001.

[24] O. Williams, A. Blake, and R. Cipolla. Sparse bayesian learning for efficient visual tracking. *IEEE PAMI*, 27(8):1292–1304, 2005.

[25] J. Winn and A. Criminisi. Object class recognition at a glance. In *Video Proc. CVPR*, 2006.

[26] J. Winn and J. Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *Proc. CVPR*, 2006.

Recognizing Actions, Objects, and Actions as Objects

Professor Mubarak Shah, School of Electrical Engineering and Computer Science, University of Central Florida, USA

Abstract

Recognition of human actions from video sequences is a very popular problem in Computer Vision. Since an action takes place in 3-D, and is projected on a sequence of 2-D images, the projected 2-D motion may vary depending on the viewpoint of the camera. This creates a problem in recognizing human actions from 2D video sequence. In most current works on *action recognition*, the issue of view-invariance has been ignored. In this talk, I will present our work on human action recognition which uses geometry to deal with the problem of view invariance.

Object recognition is a classic problem in computer vision, which has been popular in the community for the last thirty years. In the second part of my talk, I will present a novel multi-view generic object class recognition method based on 3D object modeling. Instead of using a complicated mechanism for relating multiple 2D training views, the proposed method establishes spatial connections between these views by attaching appearance features to the surfaces of 3D models. The 3D model is represented by a volume consisting of binary slices, and is generated by using a new homographic framework.

When an actor performs an action in 3D, the points on the outer boundary of the actor are projected as 2D (x, y) contour in the image plane. A sequence of such 2D contours with respect to time generates a spatiotemporal volume (STV) in (x, y, t), which can be treated as 3D object in the (x, y, t) space. In the third part of this talk, I will present our approach for human recognition by treating actions as objects. We analyze STV by using the differential geometric surface properties, such as peaks, pits, valleys and ridges, which are important action descriptors capturing both spatial and temporal properties. A set of motion descriptors for a given action is called an action sketch. The action descriptors are related to various types of motions and object deformations.

A variational framework combining level-sets and thresholding.

Samuel Dambreville, Marc Niethammer, Anthony Yezzi
and Allen Tannenbaum
School of Electrical and Computer Engineering
Georgia Institute of Technology

Abstract

Segmentation involves separating distinct regions in an image. In this note, we present a novel variational approach to perform this task within the level-sets framework. We propose an energy functional that naturally combines two segmentation techniques usually applied separately: intensity thresholding and geometric active contours. Although our method can deal with more complex statistics, we assume that the pixel intensities of the regions have Gaussian distributions, in this work. The proposed approach affords interesting properties that can lead to sensible segmentation results.

1 Introduction

Segmentation involves separating an image into distinct regions, a ubiquitous task in computer vision applications. Active contours and image thresholding are among the most important techniques for performing this task.

Geometric active contour (GAC) can easily be combined with level-set methods in which a closed curve is represented implicitly as the zero level-set of a higher dimensional function (usually a signed distance function [11]). The curve is evolved to minimize a well-chosen energy functional, typically via gradient descent (see e.g., [11, 7, 6]). The implicit representation allows the curve to naturally undergo topological changes, such as splitting and merging. Different models have been proposed to perform segmentation with GACs: Some frameworks use local image features such as edges [12, 3], whereas other methods use regional image information such a intensity statistics, color or texture [13, 1, 4, 10]. Region based approaches usually offer a higher level of robustness to noise than techniques based on local information. Many of the region-based models have been inspired by the region competition technique proposed in [15]. The book [5] is a nice general reference on the various variational segmentation methods. In region-based frameworks, the intensity statistics of the image are estimated from the segmenting curve using parametric [1, 13, 10] or non-parametric [4] methods. While nonparametric approaches allow to deal with a wide class of images, parametric methods usually result in simple, robust and efficient segmentation algorithms.

In this note, we propose a novel region-based segmentation technique with GACs. Although the proposed method is general enough to deal with diverse image statistics, we only use intensity average and variance to separate regions, in this work. Hence, our

approach is close to the parametric technique proposed in [10]. However, the way image statistics are used in our framework is different and the resulting flow exhibits distinctive properties. We define a smooth energy functional that allows to employ the result of thresholding the image in order to evolve the contour.

The literature about image thresholding is large, and a complete survey is beyond the scope of this paper. The book [2] offers a nice introduction to classical techniques about this approach. Even if a simple thresholding method is used in this work, the power of the method emanates from the combination with GACs that can naturally split or merge as well as focus on a localized portion of the image.

In what follows, we first present our method and resulting flow. Then, we report experiments, which elucidate some key aspects of this method.

2 Proposed approach: GAC and Thresholding

We consider the problem of segmenting an image $I : \Omega \mapsto \mathscr{Z}$, with $\Omega \subset \mathbf{R}^2$ and $\mathscr{Z}$ the space of possible intensity values (grey-scale). Let $x \in \Omega$ specify the coordinates of the pixels in the image I. Following [10], we assume that I is composed of two (unknown) homogeneous regions, referred to as "Object" and "Background", that have Gaussian distributions. We further assume that these regions have distinct variances[1]. The goal of segmentation is to capture these two regions. To do so, we evolve a closed curve C, represented as the zero level-set of a signed distance function $\phi : \Omega \mapsto \mathbf{R}$, such that $\phi > 0$ inside C and $\phi < 0$ outside C. Our goal is to evolve the contour C, or equivalently ϕ, so that its interior matches the Object, and its exterior matches the Background: the curve C would then match the boundary separating Object and Background.

Let us denote by $H : \mathbf{R} \mapsto \{0,1\}$ the Heaviside step function. A smooth version of H, denoted $H_\varepsilon : \mathbf{R} \mapsto [0,1]$, can be computed as follows for a chosen parameter ε:

$$H_\varepsilon(\chi) = \begin{cases} = 1 & \text{if } \chi > \varepsilon\ ; \\ = 0 & \text{if } \chi < -\varepsilon\ ; \\ = \frac{1}{2}\{1 + \frac{\chi}{\varepsilon} + \frac{1}{\pi}\sin\left(\frac{\pi\chi}{\varepsilon}\right)\} & \text{otherwise.} \end{cases} \tag{1}$$

The derivative of H_ε will be denoted by δ_ε ($\delta_\varepsilon(\chi) = 0$ if $|\chi| > \varepsilon$; and $\delta_\varepsilon(\chi) = \frac{1}{2\varepsilon}\{1 + \cos\left(\frac{\pi\chi}{\varepsilon}\right)\}$ otherwise). In this note, a region $\mathscr{R} \subset \Omega$ is characterized by a smooth labeling function $\mathrm{R}_\mathscr{R} : \Omega \mapsto [0,1]$ such as $\mathrm{R}_\mathscr{R}(x) \geq \frac{1}{2}$ if $x \in \mathscr{R}$ and $\mathrm{R}_\mathscr{R}(x) < \frac{1}{2}$ if $x \in \Omega \backslash \mathscr{R}$. Thus, the segmentation defined by the interior of the contour C is characterized by the labeling function $H_{\varepsilon_1}\phi$. [2]

The intensity averages of the pixels located inside and outside the curve C, denoted by μ_{in} and μ_{out} respectively, can be computed *at each iteration of the evolution* as

$$\mu_{\text{in}} = \frac{\int_\Omega I(x) H\phi(x)\, dx}{A_{\text{in}}} \text{ and } \mu_{\text{out}} = \frac{\int_\Omega I(x)(1 - H\phi(x))\, dx}{A_{\text{out}}}, \tag{2}$$

where $A_{\text{in}} = \int_\Omega H\phi(x)dx$ and $A_{\text{out}} = \int_\Omega \big(1 - H\phi(x)\big)dx$ denote the areas inside and outside the curve, respectively. Similarly the variances inside (σ^2_{in}) and outside (σ^2_{out}) the

[1] This is generally always the case in practice for a wide range of real-world images.

[2] For a chosen (small) ε_1. In the remainder, we will often omit ε_1 in the expressions of $H_{\varepsilon_1}\phi$, and $\delta_{\varepsilon_1}\phi$, and denote $H\phi$ and $\delta\phi$ to simplify notation.

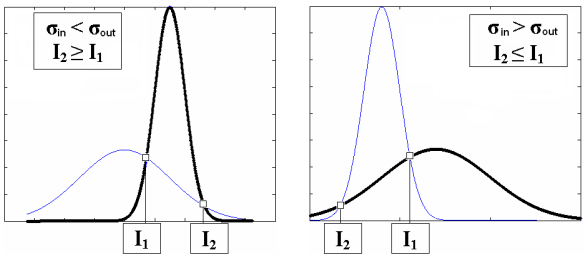


Figure 1: Probability density functions. Thick line: $P_{\text{in}}(I)$; Thin line: $P_{\text{out}}(I)$. It is straightforward to see that $P_{\text{in}}(I) > P_{\text{out}}(I)$ for $I_1 \leq I \leq I_2$, when $\sigma_{\text{in}} < \sigma_{\text{out}}$ and for $I \geq I_1$ and $I \leq I_2$, when $\sigma_{\text{in}} > \sigma_{\text{out}}$.

curve are computed as

$$\sigma_{\text{in}}^2 = \frac{\int \big(I(x) - \mu_{\text{in}}\big)^2 H\phi(x)\, dx}{A_{\text{in}}} \text{ and } \sigma_{\text{out}}^2 = \frac{\int \big(I(x) - \mu_{\text{out}}\big)^2 \big(1 - H\phi(x)\big)\, dx}{A_{\text{out}}}. \quad (3)$$

Provided a meaningful initialization of the curve, these statistics bear valuable information about the statistics of the Object and the Background. Intuitively, the intensity averages μ_{in} and μ_{out} are the best available estimates of the unknown intensity averages μ_{O} and μ_{B}, respectively. Similarly σ_{in} and σ_{out} are the best estimates of σ_{O} and σ_{B}, available at the current step of the contour evolution. We propose to use these two statistical moments to threshold the image, assuming the Gaussian distribution of pixels. From Figure 1, the (conditional) densities P_{in} and P_{out}, of the pixels belonging inside and outside the contour, respectively, are equal for two intensity values I_1 and I_2. Following the assumption $\sigma_{\text{in}} \neq \sigma_{\text{out}}$, these values can be computed as

$$\begin{aligned} I_1 &= \frac{\sigma_{\text{out}}^2 . \mu_{\text{in}} - \sigma_{\text{in}}^2 . \mu_{\text{out}} - \sigma_{\text{out}} \sigma_{\text{in}} . \alpha}{\sigma_{\text{out}}^2 - \sigma_{\text{in}}^2} \text{ and } I_2 = \frac{\sigma_{\text{out}}^2 . \mu_{\text{in}} - \sigma_{\text{in}}^2 . \mu_{\text{out}} + \sigma_{\text{out}} \sigma_{\text{in}} . \alpha}{\sigma_{\text{out}}^2 - \sigma_{\text{in}}^2}, \text{ with} \\ \alpha &= \sqrt{(\mu_{\text{in}} - \mu_{\text{out}})^2 + 2(\sigma_{\text{in}}^2 - \sigma_{\text{out}}^2) \log(\frac{\sigma_{\text{in}}}{\sigma_{\text{out}}})}. \end{aligned} \quad (4)$$

To perform the thresholding operation, we define the labeling function $G : \Omega \mapsto [0,1]$ so that pixels x that are more likely to belong to the Object, i.e., $P_{\text{in}}\big(I(x)\big) \geq P_{\text{out}}\big(I(x)\big)$, are assigned a value $G(x) \geq \frac{1}{2}$, whereas pixels more likely to belong to the Background, i.e., $P_{\text{in}}\big(I(x)\big) < P_{\text{out}}\big(I(x)\big)$, are assigned a value $G(x) < \frac{1}{2}$. Based on Figure 1, a natural candidate for the function G is

$$\text{if } \sigma_{\text{in}} < \sigma_{\text{out}},\ G(x) = \begin{cases} \geq \frac{1}{2} & \text{if } \ I_2 \geq I(x) \geq I_1\ ; \\ < \frac{1}{2} & \text{else}\ . \end{cases}$$

$$\text{if } \sigma_{\text{in}} > \sigma_{\text{out}},\ G(x) = \begin{cases} \geq \frac{1}{2} & \text{if } \ I(x) \geq I_1 \text{ or } I(x) < I_2\ ; \\ < \frac{1}{2} & \text{else}\ . \end{cases}$$

Using the smooth version of the Heaviside function described above (for a chosen smooth-

ness parameter ε_2), the function G can be chosen to be[3]

$$G(x) = \begin{cases} H_{\varepsilon_2}\big(I(x)-I_1\big) - H_{\varepsilon_2}\big(I(x)-I_2\big) & \text{if } \sigma_{\text{in}} < \sigma_{\text{out}} \text{ ;} \\ 1 + H_{\varepsilon_2}\big(I(x)-I_1\big) - H_{\varepsilon_2}\big(I(x)-I_2\big) & \text{if } \sigma_{\text{in}} > \sigma_{\text{out}} \end{cases} \tag{5}$$

To perform the segmentation, we propose to minimize

$$E_{\text{image}}(\phi, I) = \|H\phi - G\|^2 = \int_\Omega \{H\phi(x) - G(x)\}^2 dx. \tag{6}$$

This energy is the L_2 distance between the (*current*) segmentation $H\phi$ obtained from the curve C and the (*implied*) segmentation G obtained from thresholding the image I. The minimization is performed by evolving C according to the flow:

$$\frac{d\phi}{dt} = -\nabla_\phi E_{\text{image}} + \lambda.\delta\phi.\text{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) \tag{7}$$

where the second term in the right-hand side is a regularizing term penalizing high curvatures (the parameter $\lambda \in \mathbf{R}^+$ is chosen subjectively). The gradient $\nabla_\phi E_{\text{image}}$ can be computed using calculus of variations (both $H\phi$ and G depend on ϕ):

$$\nabla_\phi E_{\text{image}} = 2\delta\phi\big(H\phi - G\big) + 2\big(\beta_1.\nabla_\phi I_1 - \beta_2.\nabla_\phi I_2\big) \tag{8}$$

where the expressions of $\beta_1 \in \mathbf{R}$ and $\beta_2 \in \mathbf{R}$ are

$$\beta_1 = \int_\Omega \delta_{\varepsilon_2}\big(I(x)-I_1\big)[H\phi(x)-G(x)]dx \;\&\; \beta_2 = \int_\Omega \delta_{\varepsilon_2}\big(I(x)-I_2\big)[H\phi(x)-G(x)]dx. \tag{9}$$

The expressions of $\nabla_\phi I_1$ and $\nabla_\phi I_2$ are detailed in the Appendix. The gradient $\nabla_\phi E_{\text{image}}$ is the sum of two terms. The first term, namely $2\delta\phi\big(H\phi - G\big)$, simply points towards G: Image pixels (in the vicinity of C), which are more likely to belong to the Object are included in C, whereas pixels more likely to belong to the Background are excluded. Hence, this term influences the contour based only on statistical considerations. As will be highlighted in the experimental part, this first term drives the contour in a similar fashion as the gradient proposed in [10], which also stems from statistical considerations of the intensity distributions only. The second term, namely $2\big(\beta_1.\nabla_\phi I_1 - \beta_2.\nabla_\phi I_2\big)$, involves the gradients of the thresholds I_1 and I_2 with respect to ϕ (which in turn involve $\nabla_\phi \mu_{\text{in}}$, $\nabla_\phi \mu_{\text{out}}$, $\nabla_\phi \sigma_{\text{in}}$ and $\nabla_\phi \sigma_{\text{out}}$; see Appendix). Such a term is not present in the flows presented in [1, 10]. Depending on the sign of β_1 (respectively, β_2), this second term points in the direction of the fastest increase or decrease of the threshold I_1 (respectively, I_2), *resulting in changing G to decrease* E_{image}. The coefficients β_1 and β_2 are essentially counting functions focusing on the "ambiguous pixels", i.e. pixels which intensities are close to the thresholds I_1 and I_2 (C.f. the terms $\delta(I-I_1)$ and $\delta(I-I_2)$, in the expressions of β_1 and β_2) and for which classification is the most uncertain. The sign and magnitude of β_1 and β_2 are conditioned upon the number of ambiguous pixels that belong to one of the segmentations ($H\phi$ or G) but not the other. To better understand the influence of this second term, we propose to consider the following example:

[3] In the limit case, when $\varepsilon_2 \to 0$, G is the result of *hard thresholding* the image I using the band(s) defined by I_1 and I_2.

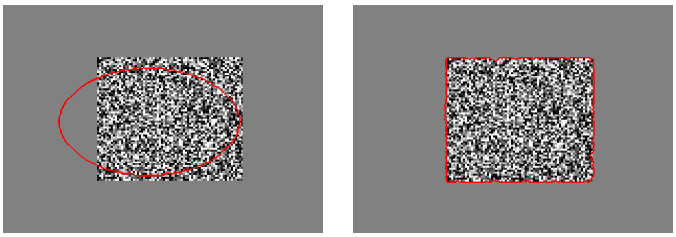


Figure 2: Synthetic Example, two regions of same mean intensity and different variances. *Left*: Initialization ; *Right*: Final segmentation.

Let us assume that ambiguous pixels with values close to I_1 are in majority outside the contour. The coefficient β_1 is thus negative. In the gradient descent process of Equation (7), the curve evolves in the direction of $\nabla_\phi I_1$, which increases I_1. This results in excluding ambiguous pixels from the segmentation G leading to a better match with the segmentation defined by the curve.
Hence, the second term of $\nabla_\phi E_{\text{image}}$ appears to make meaningful decisions of the type: if more ambiguous pixels are inside C (resp. outside) then take more ambiguous pixels inside G (resp. outside). This systematic treatment of the thresholds I_1 and I_2 afforded by the second term of $\nabla_\phi E_{\text{image}}$ is an important feature of this work, and will be further examined in the experimental part below.

3 Experimental Results

We now present experimental results, typically using the heuristic $\varepsilon_2 = 3 \times \varepsilon_1$ for gray-scale images $I : \Omega \mapsto \{0..255\}$. The parameter λ was chosen subjectively in $[0,2]$.

In Figure 2, a synthetic image composed of two regions that have the same average intensities and different variances is successfully segmented with our method.

In Figure 3, two textured natural images are segmented. The segmentation of the leopard obtained with our method is convincing compared to the segmentation obtained in [4], where the mutual information between general distributions of pixel intensities was minimized. The zebra is also successfully segmented even though only the average and variance of pixel intensities are used (This image was segmented before using more involved image information, such as texture [8] or general pixel distribution [4, 9]).

Segmenting medical images is usually a challenging task, since structures of interest are often poorly contrasted with respect to other neighboring structures. The proposed method was found to perform in a satisfactory manner on such images. In what follows, we report three experiments and contrast our results with results obtained using the method proposed in [10] that uses the same image information (intensity averages and variances). In Figure 4, the hand of a patient with a Kaposi Sarcoma (KS) is segmented. The contour is initialized in the vicinity of the KS, and an acceptable segmentation of the pathology is obtained with our method (i.e., using the full expression of ∇E_{image} as in Equation (8)). Using the flow proposed in [10], for the same initialization, the contour ends up capturing the whole visible part of the hand. A similar result was obtained using the first term in ∇E_{image} only. In Figures 5 and 6, MRI images of the heart are segmented.

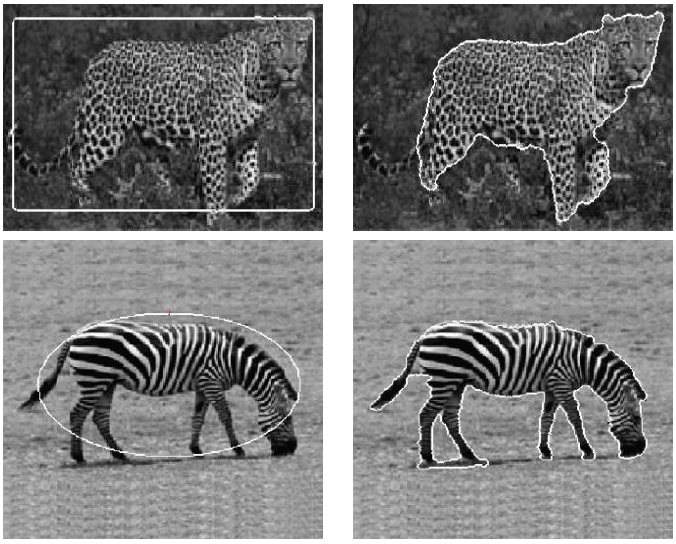
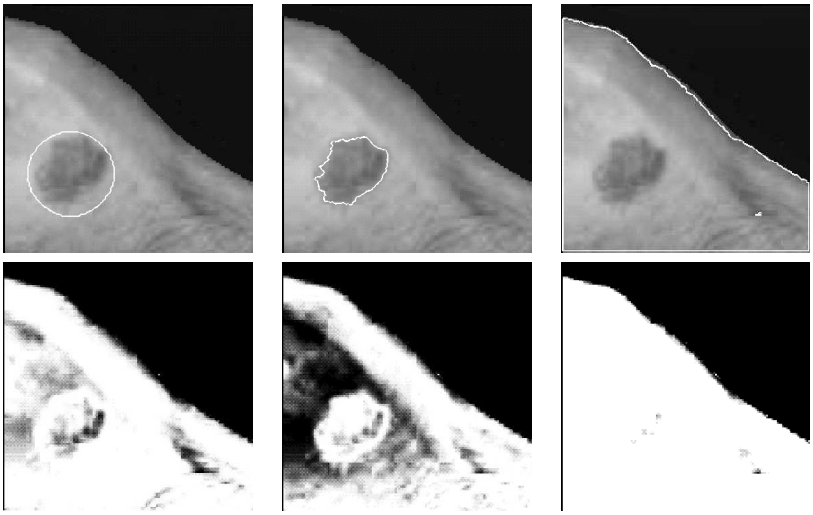


Figure 3: Segmentation results of two natural images. *Left*: Initialization, *Right*: Segmentation result with the proposed method.

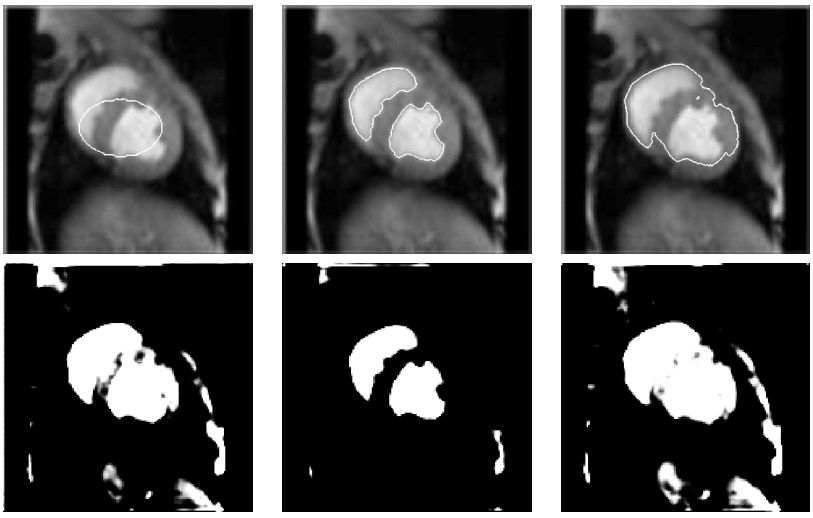
Convincing segmentations of the left ventricle are obtained with our method (The results obtained were identical for a wide range of shapes of the initial contour). A comparable segmentation of the image in Figure 5 was obtained in [14], where global constraints were imposed on the flow presented in [13]. Using the method proposed in [10] or the first term of ∇E_{image} only, the ventricle is not properly segmented. In Figure 7, a transverse MRI-image of the brain is presented. The caudate nuclei are notoriously difficult to segment due to their poor contrast with neighboring structures. An acceptable segmentation of the left and right caudate is obtained with our method. Using the method proposed in [10] or the first term of ∇E_{image} only, the whole white and grey matter of the brain is segmented, even though the contour was initialized in the vicinity of the caudate nuclei.

Hence, in view of the experiments performed, our method appears to lead to meaningful segmentations notably in the case of medical images. When only the first term of ∇E_{image} is used, similar results as [10] were obtained. This highlights the influence of the second term in ∇E_{image}. This term evolves the contour in ways that modify G to match $H\phi$ as fast as possible (by changing the statistics inside and outside the curve). This can be further noticed from observing the final thresholded images in Figures 4 and 5, whether the second term of ∇E_{image} is used or not. Thus, the second term in ∇E_{image} has the advantageous effect of better exploiting some of the information built in the initial contour (e.g., placement, shape or size) and allows for the segmentation of objects/structures that are in its vicinity. Furthermore, these experiments suggest that using statistical information alone may not be enough to lead to meaningful segmentation results.



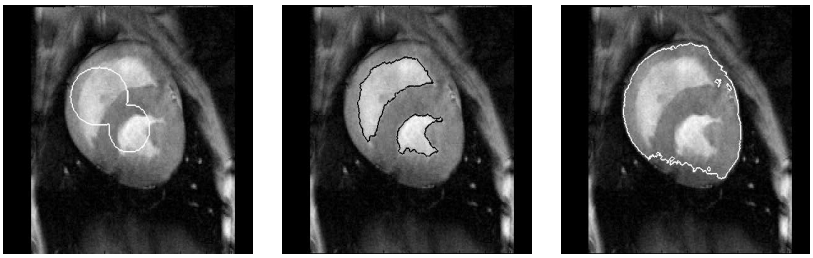
(a) (b) (c)

Figure 4: Segmentation of the hand of a patient with a Kaposi Sarcoma. *Top Row*: (a)-Initialization; (b)-Final segmentation with the proposed method; (c)-Final segmentation with the method proposed in [10] or with the first term of ∇E_{image} only. *Bottom Row*: Corresponding thresholded images.



(a) (b) (c)

Figure 5: Segmentation of an MRI image of the heart. *Top Row*: (a)-Initialization; (b)-Final segmentation with the proposed method; (c)-Final segmentation with the method proposed in [10] or with the first term of ∇E_{image} only. *Bottom Row*: Corresponding thresholded images.



(a) (b) (c)

Figure 6: Other segmentation of an MRI image of the heart. (a)-Initialization; (b)-Final segmentation with the proposed method; (c)-Final segmentation with the method proposed in [10] or with the first term of ∇E_{image} only.

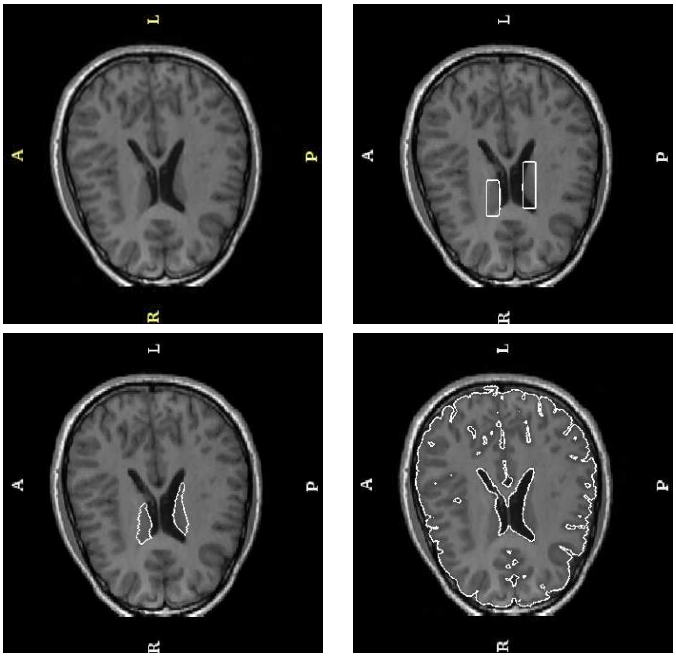


Figure 7: Segmentation of the caudate nuclei. *Top Left*: MRI image of the brain. *Top Right*: Contour initialization. *Bottom Left*: Final segmentation with the proposed method. *Bottom Right*: Final segmentation with the method proposed in [10] or with the first term of ∇E_{image} only.

4 Conclusions and Future Research

In this work, we presented a variational approach combining two segmentation techniques: active contours and thresholding. In the proposed framework, image statistics (intensity averages and variances) are exploited in a novel manner by defining an energy functional that compares the segmentation obtained from the evolving curve with the segmentation obtained from thresholding the image. Our region-based approach was shown to lead to convincing results on nontrivial real-world examples that were successfully segmented before, but using much more involved segmentation schemes. The proposed method also led to improved results for a variety of medical images in comparison with a previously proposed variational approach that uses the same statistical information. The flow resulting from our energy functional seems to afford some useful properties allowing the segmenting curve to reach local energy minima that coincide with meaningful segmentations. In our future work, we plan to expand the proposed approach to incorporate various image information such as intensity histograms or texture. The use and influence of different metrics to define our energy is also the interest of our future research.

References

[1] T. Chan and L. Vese. Active contours without edges. *IEEE Trans. on Image Processing*, 10(2):266–277, 2001.

[2] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Addison-Wesley Longman., 2001.

[3] S. Kichenassamy, S. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Conformal curvature flow: From phase transitions to active vision. In *Archives for Rational Mechanics and Analysis*, volume 134, pages 275–301, 1996.

[4] J. Kim, J. Fisher, A. Yezzi, M. Cetin, and A. Willsky. Nonparametric methods for image segmentation using information theory and curve evolution. In *Proc. ICIP*, volume 3, pages 797–800, 2002.

[5] J-M Morel and S. Solimini. *Variational Methods for Image Segmentation*. Birkhauser, 1994.

[6] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer Verlag, 2003.

[7] N Paragios, Y. Chen, and O Faugeras. *Handbook of Mathematical Models in Computer Vision*. Springer, 2005.

[8] N. Paragios and R. Deriche. Geodesic active regions for supervised texture segmentation. In *ICCV (2)*, pages 926–932, 1999.

[9] Y. Rathi, O. Michailovich, and A. Tannenbaum. Seeing the unseen: Segmenting with distributions. In *Intl. Conf. Signal and Image Processing*, volume 534, 2006.

[10] M. Rousson and R. Deriche. A variational framework for active and adaptative segmentation of vector valued images. In *Proc. of the Workshop on Motion and Video Computing*, page 56, 2002.

[11] J. A. Sethian. *Level Set Methods and Fast Marching Methods*. 1999.

[12] V.Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *IJCV*, volume 22, pages 61–79, 1997.

[13] A. Yezzi, A. Tsai, and A. Willsky. A statistical approach to snakes for bimodal and trimodal imagery. In *Proc. ICCV*, volume 2, pages 898–903, 1999.

[14] A. Yezzi, A. Tsai, and A. Willsky. Medical image segmentation via coupled curve evolution equations with global constraints. In *Proc. Workshop on Mathematical Methods in Biomedical Image Analysis*, pages 12–19, 2000.

[15] Song Chun Zhu and Alan L. Yuille. Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. volume 18, pages 884–900, 1996.

5 Appendix: Further details on the proposed flow

We now detail the expressions of $\nabla_\phi I_1$ and $\nabla_\phi I_2$, which appear in Equation (8). These expressions involve $\nabla_\phi \mu_{\text{in}}$, $\nabla_\phi \mu_{\text{out}}$, $\nabla_\phi \sigma_{\text{in}}$ and $\nabla_\phi \sigma_{\text{out}}$, which are given by

$$\begin{aligned} \nabla_\phi \mu_{\text{in}} &= \delta\phi \left(\frac{I-\mu_{\text{in}}}{A_{\text{in}}} \right) \quad , \quad \nabla_\phi \mu_{\text{out}} = \delta\phi \left(\frac{\mu_{\text{out}}-I}{A_{\text{out}}} \right) \\ \nabla_\phi \sigma_{\text{in}} &= \delta\phi \left(\frac{(I-\mu_{\text{in}})^2-\sigma_{\text{in}}^2}{2\sigma_{\text{in}}.A_{\text{in}}} \right) \quad \text{and} \quad \nabla_\phi \sigma_{\text{out}} = \delta\phi \left(\frac{\sigma_{\text{out}}^2-(I-\mu_{\text{out}})^2}{2\sigma_{\text{out}}.A_{\text{out}}} \right). \end{aligned} \tag{10}$$

From Equation (4) and using classical rules of calculus, one gets

$$\begin{aligned} \nabla_\phi I_1 &= m_1^{\text{in}}.\nabla_\phi \mu_{\text{in}} + m_1^{\text{out}}.\nabla_\phi \mu_{\text{out}} + s_1^{\text{in}}.\nabla_\phi \sigma_{\text{in}} + s_1^{\text{out}}.\nabla_\phi \sigma_{\text{out}}. \\ \nabla_\phi I_2 &= m_2^{\text{in}}.\nabla_\phi \mu_{\text{in}} + m_2^{\text{out}}.\nabla_\phi \mu_{\text{out}} + s_2^{\text{in}}.\nabla_\phi \sigma_{\text{in}} + s_2^{\text{out}}.\nabla_\phi \sigma_{\text{out}}. \end{aligned} \tag{11}$$

where the parameters m_1^{in}, m_1^{out}, s_1^{in} and s_1^{out} are

$$\begin{aligned} m_1^{\text{in}} &= \frac{1}{\sigma_{\text{out}}^2-\sigma_{\text{in}}^2} \left(\sigma_{\text{out}}^2 - \sigma_{\text{in}}\sigma_{\text{out}} \left(\frac{\mu_{\text{in}}-\mu_{\text{out}}}{\alpha} \right) \right) \\ m_1^{\text{out}} &= \frac{1}{\sigma_{\text{out}}^2-\sigma_{\text{in}}^2} \left(\sigma_{\text{in}}\sigma_{\text{out}} \left(\frac{\mu_{\text{in}}-\mu_{\text{out}}}{\alpha} \right) - \sigma_{\text{in}}^2 \right) \\ s_1^{\text{in}} &= \frac{1}{\sigma_{\text{out}}^2-\sigma_{\text{in}}^2} \Big\{ 2\sigma_{\text{in}}\sigma_{\text{out}} \left(\frac{\sigma_{\text{out}}\mu_{\text{in}} - \sigma_{\text{out}}\mu_{\text{out}} - \sigma_{\text{in}}.\alpha}{\sigma_{\text{out}}^2-\sigma_{\text{in}}^2} \right) - \\ &\qquad \left(\sigma_{\text{out}}.\alpha + \frac{2\sigma_{\text{in}}^2\sigma_{\text{out}}}{\alpha} \log \left(\frac{\sigma_{\text{in}}}{\sigma_{\text{out}}} \right) + \sigma_{\text{out}} \frac{\sigma_{\text{in}}^2-\sigma_{\text{out}}^2}{\alpha} \right) \Big\} \\ s_1^{\text{out}} &= \frac{1}{\sigma_{\text{out}}^2-\sigma_{\text{in}}^2} \Big\{ 2\sigma_{\text{in}}\sigma_{\text{out}} \left(\frac{\sigma_{\text{in}}\mu_{\text{out}} - \sigma_{\text{in}}\mu_{\text{in}} + \sigma_{\text{out}}.\alpha}{\sigma_{\text{out}}^2-\sigma_{\text{in}}^2} \right) - \\ &\qquad \left(\sigma_{\text{in}}.\alpha - \frac{2\sigma_{\text{in}}\sigma_{\text{out}}^2}{\alpha} \log \left(\frac{\sigma_{\text{in}}}{\sigma_{\text{out}}} \right) - \sigma_{\text{in}} \frac{\sigma_{\text{in}}^2-\sigma_{\text{out}}^2}{\alpha} \right) \Big\} \end{aligned}$$

The expressions for the parameters m_2^{in}, m_2^{out}, s_2^{in} and s_2^{out} are identical to the expressions of m_1^{in}, m_1^{out}, s_1^{in} and s_1^{out}, replacing α by $-\alpha$.

Automatic 3D Object Segmentation in Multiple Views using Volumetric Graph-Cuts

Neill Campbell[1] George Vogiatzis[2] Carlos Hernández[2] Roberto Cipolla[1]

[1]Department of Engineering, University of Cambridge, Cambridge, CB2 1PZ, UK

[2]Toshiba Research Europe, 208 Cambridge Science Park, Milton Road, Cambridge, CB4 0GZ, UK

{ndfc2, cipolla}@eng.cam.ac.uk {george.vogiatzis, carlos.hernandez}@crl.toshiba.co.uk

Abstract

We propose an algorithm for automatically obtaining a segmentation of a rigid object in a sequence of images that are calibrated for camera pose and intrinsic parameters. Until recently, the best segmentation results have been obtained by interactive methods that require manual labelling of image regions. Our method requires no user input but instead relies on the camera *fixating* on the object of interest during the sequence. We begin by learning a model of the object's colour, from the image pixels around the fixation points. We then extract image edges and combine these with the object colour information in a volumetric binary MRF model. The globally optimal segmentation of 3D space is obtained by a graph-cut optimisation. From this segmentation an improved colour model is extracted and the whole process is iterated until convergence.

Our first finding is that the fixation constraint, which requires that the object of interest is more or less central in the image, is enough to determine what to segment and initialise an automatic segmentation process. Second, we find that by performing a single segmentation in 3D, we implicitly exploit a 3D rigidity constraint, expressed as *silhouette coherency*, which significantly improves silhouette quality over independent 2D segmentations. We demonstrate the validity of our approach by providing segmentation results on real sequences.

1 Introduction

Recently there has been growing interest in the development of dense stereo reconstruction techniques [10]. These techniques focus on producing 3D models from a sequence of calibrated images of an object where the intrinsic parameters and pose of the cameras are known. In addition to the multiple views, many of these techniques also require knowledge of the object's silhouette in each view to provide: (a) an approximate initial reconstruction, (b) an outer bound for the reconstructed object and (c) approximate occlusion reasoning [12]. This demand for accurate object silhouettes will influence the acquisition of the image sequence to provide an easy segmentation task, for example sequences are taken on turntables against fixed coloured or known backgrounds. Naturally this places limitations on the subject matter for reconstruction since real world objects

will often be immovable and there will be no control over the image background which may well vary considerably with differing views of the object. To extend the applicability of multi-view stereo to real world objects it is necessary to be able to segment a rigid object in a sequence of possibly cluttered images.

The most recent advances in image segmentation adopt an interactive approach [2, 3, 9] where the user is required to guide the segmentation by manually segmenting image regions. The information supplied by the user is used to learn the object's colour distribution and the globally optimal segmentation is achieved by the maximum flow algorithm [7]. This approach performs well on simple images, but the extent of user interaction required increases significantly if the object contains multiple distinct colours and if it is found in a cluttered or camouflaged environment.

If we now consider the case of a large number of images, required for high accuracy reconstructions, we can see that even a modest amount of user interaction on an individual image basis will represent a significant task using these interactive tools. In order to reduce the demands placed on the user we aim to exploit the constraints that exist within the image sequence. Note that the sequence contains a series of multiple views of the same rigid 3D object, therefore the segmentations must satisfy a *silhouette coherency* constraint [6]. This constraint follows from the knowledge that the images are formed from projections of the same rigid 3D object with the correct segmentations being the corresponding silhouettes. This suggests that by attempting to perform segmentation on isolated 2D images we are rejecting a great deal of information contained within the sequence. We further observe that, since the intended purpose of the image sequence is to reconstruct the object, we have a *fixation* constraint whereby the object of interest is always being focused upon by the camera and thus likely to be central to the viewpoint.

In order to exploit the coherency constraint we perform the segmentation across all images simultaneously. This is achieved by performing a binary segmentation of 3D space where each 3D location (voxel) is labelled as 'object' or 'background'. The 3D segmentation is influenced by: (a) a colour model initially learned from the fixation points and then refined in subsequent iterations and (b) a 3D shape prior based on minimal surface area.

The advantages of our approach over independent 2D interactive segmentations are twofold. Firstly, we replace the user supplied image labelling with the fixation constraint, which is usually satisfied by multi-view stereo sequences. Additionally we employ constraints that link multiple segmentations of the same rigid object to improve segmentation quality.

The rest of the paper is laid out as follows: In Section 2 we review relevant prior work and discuss the differences of our approach. Section 3 begins with an overview of the segmentation algorithm and continues to provide details of the automatic initialisation procedure. A discussion of how the colour models are learnt follows in Section 4 and Section 5 establishes the volumetric graph-cut framework used to perform the optimisation. In Section 6 we provide two sets of results of real world image sequences. The first shows the complete reconstruction of a hat in a fully automatic process with the user only providing the input images. The second provides a much more challenging segmentation task and we compare the performance of our volumetric approach against automatic 2D segmentations. The paper concludes in Section 7 where the main contributions of the paper are reviewed.

2 Previous Work

The work of [3] introduces a framework for adopting an energy minimisation approach to segmentation where the resulting optimisation process is both tractable and unique thanks to the max-flow/min-cut algorithm. Although the paper details how the method may be extended to N-Dimensional graphs, the predominant focus is on the segmentation of 2D images. They also demonstrate a simple histogram based learning process to allow a user to interactively segment a grayscale image. This approach was formalised in [9] where an interactive system for the segmentation of 2D colour images was proposed. As with our method, that paper adopts an iterative approach to the segmentation task by learning colour models of the object and the background. Although considered state-of-the-art for 2D images, the interactive demands placed on the user make the segmentation of a large sequence of images a sizeable task which we avoid by adopting an automatic approach.

An extension of this idea is presented in [13], where a deforming object is segmented in a video sequence by a graph-cut on a 3D 'space-time' volume. Although related, the approach of [13] is not directly applicable to our problem since our image sequence is not obtained from video and as a result is not dense enough to form a continuous space-time volume.

In [11], a graph-cut based approach is used to estimate the voxel occupancy of a calibrated volume in space. Their approach is directly aimed at using an energy minimisation framework to regularise the process of combining a series of imperfect silhouettes. The main difference is that they obtain these silhouettes as the result of a background subtraction process from a fixed, calibrated camera rig whereas we adopt an iterative learning approach requiring no prior knowledge of the object or environment.

The task of segmenting objects in multi-views has also been studied in [14]. Whilst also approaching the task in the 3D domain, here the authors use a level set method to evaluate the segmentation based on Lambertian scenes with smooth of constant albedo. Level set methods are known to be susceptible to local minima so [14] relies on smooth albedo variation and a multi-resolution scheme to achieve convergence. In contrast, our method tolerates albedo discontinuities and, due to the graph-cut optimisation scheme, is guaranteed to converge to the globally optimal segmentation.

3 Problem Statement

We accept as an input a sequence of M images, $I_1 ... I_M$, of the object with each image made up of a set of pixels $\mathcal{P}_m$. We assume the images are calibrated which allows any location in the voxel volume, $\mathbf{X} \in \mathbb{R}^3$, to be mapped to its corresponding location in image I_m, $\mathbf{x}_m \in \mathbb{R}^2$. We also assume our fixation condition where the object is taken to be fully contained by and centred in each view. We form an array of N voxels, $\mathcal{V}$, from the intersection of the volumes projected by each of the images, thus the fixation constraint mandates that this volume contain the visual hull of the object. Each voxel has dimensions $(\delta x, \delta y, \delta z)$ and may be indexed as $v_n \in \mathcal{V}$. We intend to label each voxel as inside the object's visual hull ($\mathcal{O} \subset \mathcal{V}$) or outside ($\mathcal{B} \subset \mathcal{V}$), and thus part of the background in the image segmentation, such that $v_n \in \mathcal{O} \cup \mathcal{B}$. We also define $\mathbf{u}_{m,n} \in \mathcal{P}_m$ where $\mathbf{u}_{m,n} = I_m(\mathbf{Proj}(v_n))$ is the RGB colour of the pixel which v_n projects to in image I_m. When formulating the voxel array as a graph we define a set of edges $\mathcal{E}$ containing neighbouring voxels in a six-connected sense.

```
Acquire calibrated image sequence fixating on the object
Generate voxel array from bounding volume
Use fixation points to initialise colour models
Use graph-cut to produce initial segmentation
while Visual hull not converged do
    Extract the silhouettes of the current visual hull segmentation
    Update the object colour model using all images
    Update background colour model for each image
    Generate classification probabilities
    Update graph with new values
    Perform graph-cut segmentation to produce new visual hull
end while
```

Figure 1: The iterative segmentation algorithm.

An overview of our segmentation algorithm is given in Figure 1. The first stage of the algorithm is to establish a location within the voxel array known to be contained within the visual hull of the object. This is required to provide a starting point for the estimation of the colour model of the object. We make use of the fixation condition previously mentioned to automatically generate this seed location from the centroid of intersection of the optical axes of the individual cameras and sample with a Gaussian weighting, of conservative radius, to provide the data for the initial object colour model. The initialisation is used to provide the starting point for the second stage of the algorithm, namely the iterative refinement of the visual hull estimation using a volumetric graph-cut scheme. The following two sections discuss the main components of the iterative stage: building the colour models and the volumetric graph-cut.

4 Building Colour Models

During the algorithm we develop colour models to provide probabilistic likelihoods for image pixels to be part of the object or background. In common with general practice in this area [9] we use a K component Gaussian Mixture Model (GMM), [1], in 3D colour space (red, green, blue) to model the likelihood. These models take the form of (1) where $\mathbf{u}$ is a vector in colour space. The probability distribution of the colour of the k^{th} component of the mixture is given by a normal distribution with mean μ_k and covariance Σ_k. Each of the individual components is weighted by the marginal probability of the component $p(k)$, termed the mixing coefficient and denoted π_k. The number of mixture components, K, provides a trade-off between computation time and model over-fitting against the discriminative power of the colour model. For our experiments we used $K = 5$.

$$p(\mathbf{u} \,|\, \pi_k, \mu_k, \Sigma_k) \;=\; \sum_{k=1}^{K} p(k)\, p(\mathbf{u} \,|\, \mu_k, \Sigma_k) \;=\; \sum_{k=1}^{K} \pi_k \,\mathcal{N}(\mathbf{u} \,|\, \mu_k, \Sigma_k) \qquad (1)$$

At each iteration, GMMs are learnt for both the object and the background. The learning process consists of sampling the pixels as a sequence of colour vectors and using the Expectation-Maximisation (EM) algorithm to 'fit' the model parameters of the GMM to the sampled data [1]. We intend to exploit the fact that the object is seen in multiple

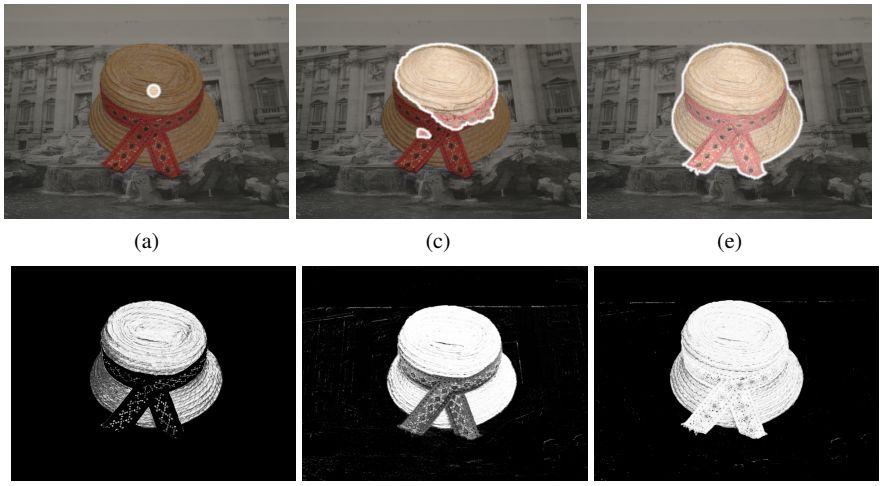


Figure 2: Iterative learning of the object colour model. *The fixation condition is used to provide a seed (a) for the first object colour model (b). In all images the seed areas contain only the straw colour. The result of the first graph-cut (c) contains areas of the red ribbon which is incorporated into the colour model (d). The second iteration refines this to produce the correct segmentation (e) with corresponding colour model (f). (b,d,e) Show the likelihood of being object* $\hat{\mathrm{L}}_{\mathcal{O}}$ *under the colour model at each iteration (7).*

views, therefore we build a full colour model for the object by sampling pixels from all the views using the silhouettes as a mask. This approach allows us to increase our knowledge about all the colours present in the object even if the initialisation fails to capture all the colours of the object. In this situation an automatic 2D segmentation would most likely fail. However, under our method the visual hull estimation forces a spatial coherency, thus when the graph-cut is performed the segmentation will generate a visual hull whose silhouettes should include a portion of the colour in at least one of the views. Since the colour model is built over all views, we only require one view to register the second colour as object in order to allow the subsequent iterations to add the colour to the model and extend the volume into the other region.

This is demonstrated in the hat sequence of Figure 2. The initialisation of the colour model, Figure 2 (a), contained only the straw colour in each view, therefore the first iteration object colour model, Figure 2 (b), fails to classify the red ribbon as object. The result of the first graph-cut segmentation, given in Figure 2 (c), attempts to combine the separated straw coloured regions and in the process includes some of the ribbon in the segmentation. The second iteration learns an object model which propagates this knowledge of the ribbon over all the image sequences so that the object colour model includes the ribbon as in Figure 2 (d). This is refined in the subsequent iteration to produce the correct segmentation of Figure 2 (e), with the object colour model Figure 2 (f), after the second iteration graph-cut.

In contrast to the object model, we expect the background model to vary over each image, therefore a separate GMM is learnt for the background of each image. Since we are unsure of the final visual hull at the start of the iteration, we conservatively estimate the background by sampling image pixels which aren't currently segmented as the object.

In the early stages, particularly during the first iteration, the sampled data will contain many pixels from the object itself. However, the localised object model will necessarily be a better fit to the data in and around the seed location, therefore the more generalised background model will not prevent the initial region from growing.

Since the algorithm is EM based it is possible that convergence will be to a local rather than global optimum. However, since we reject background pixels using spatial consistency, the algorithm is forced to converge on a spatially coherent object.

5 Volumetric Graph-Cut

The task of segmentation is performed within the voxel array and the resulting silhouettes from the computed visual hull propagated back to the individual images. This ensures the set of image silhouettes are consistent with one another at every iteration. The segmentation operation is one of energy minimisation as in (2). The energy to be minimised, given in (3), is comprised of two terms: a volumetric term and a boundary term. The parameter Θ denotes the collection of colour model parameters , as in (4), which are updated at each iteration using the results of the previous segmentation.

$$\mathcal{O} = \arg\min_{\mathcal{O} \subset \mathcal{V}} \mathrm{E}(\mathcal{O}, \mathcal{B}, \{I_m\}, \Theta) \tag{2}$$

$$\mathrm{E}(\mathcal{O}, \mathcal{B}, \{I_m\}, \Theta) = \lambda\, \mathrm{E}_{\mathrm{vol}}(\mathcal{O}, \mathcal{B}, \{I_m\}, \Theta) + (1-\lambda)\, \mathrm{E}_{\mathrm{surf}}(\mathcal{O}, \mathcal{B}, \{I_m\}) \tag{3}$$

$$\text{where} \quad \Theta = \Big[\{\pi_k^{\mathcal{O}}\}, \{\mu_k^{\mathcal{O}}\}, \{\Sigma_k^{\mathcal{O}}\}, \{\pi_{k,m}^{\mathcal{B}}\}, \{\mu_{k,m}^{\mathcal{B}}\}, \{\Sigma_{k,m}^{\mathcal{B}}\} \Big] \tag{4}$$

5.1 Volume Term

The volume term encodes the preference for a voxel to be classified as inside or outside the object. We therefore construct this term from the colour models of the individual views. For the m^{th} image in the sequence we can evaluate a likelihood term for the projected pixel colour $\mathbf{u}_{m,n}$ of voxel v_n to be part of the object (5) or the background (6) given the current GMM model parameters in the same form as (1).

$$\mathrm{L}_{\mathcal{O}}(v_n, \Theta, m) = p(\mathbf{u}_{m,n} \,|\, \pi_k^{\mathcal{O}}, \mu_k^{\mathcal{O}}, \Sigma_k^{\mathcal{O}}) \tag{5}$$

$$\mathrm{L}_{\mathcal{B}}(v_n, \Theta, m) = p(\mathbf{u}_{m,n} \,|\, \pi_{k,m}^{\mathcal{B}}, \mu_{k,m}^{\mathcal{B}}, \Sigma_{k,m}^{\mathcal{B}}) \tag{6}$$

In the absence of any prior knowledge about the pixels class, we may form a classification probability of being object by normalising the likelihoods. We then sum over all images for a single voxel, normalising by the number of images, as shown by (7).

$$\hat{\mathrm{L}}_{\mathcal{O}}(v_n, \Theta) = \frac{1}{M} \sum_{m=1}^{M} \frac{\mathrm{L}_{\mathcal{O}}(v_n, \Theta, m)}{\mathrm{L}_{\mathcal{O}}(v_n, \Theta, m) + \mathrm{L}_{\mathcal{B}}(v_n, \Theta, m)} \tag{7}$$

The final volume cost is given by (8). Rather than use a per-image binary voting cost [11] we combine the probabilities from the individual images using an offset parameter $\phi \in [0, 1]$. This parameter encodes the threshold level of the algorithm — were we simply to combine binary silhouettes it would represent the number of images which have to

agree in order to start considering a particular voxel as part of the visual hull as seen in the case of ideal silhouettes where we have $\phi \rightarrow 1$ since $\hat{\mathrm{L}}_{\mathcal{O}}(v_n, \Theta) \rightarrow 1$ in the event of perfect object classification in all images. Thus ϕ controls the rate of convergence to the true solution against robustness to noise in the event of imperfect colour model classification in each image. Usually a conservative value of around 0.8 - 0.9 results in the best trade-off.

$$\mathrm{E}_{\mathrm{vol}}(\mathcal{O}, \mathcal{B}, \{I_m\}, \Theta) \quad = \quad \sum_{v_n \in \mathcal{V}} \begin{cases} \left(1 - \left[\hat{\mathrm{L}}_{\mathcal{O}}(v_n, \Theta) - \phi\right]\right) & v_n \in \mathcal{O} \\ \left(1 + \left[\hat{\mathrm{L}}_{\mathcal{O}}(v_n, \Theta) - \phi\right]\right) & v_n \in \mathcal{B} \end{cases} \tag{8}$$

5.2 Boundary Term

The boundary term in (3) encodes the energy associated with the surface area of the object. In a similar manner to the 2D examples of [3, 9], we use the colour discontinuities within the images to project cutting planes through the voxel array which should form the boundaries of the visual hull. We identify the colour difference as the vector norm of the projected pixels of neighbouring voxels in each image as (9).

$$Z_m(v_i, v_j) \quad = \quad ||\mathbf{u}_{m,i} - \mathbf{u}_{m,j}||^2 \tag{9}$$

Since we are estimating the boundary of the visual hull, the maximum colour difference across the image sequence is taken and an exponential cost function, with β estimated from the images [3, 9], used to provide the standard Gibb's model of (10).

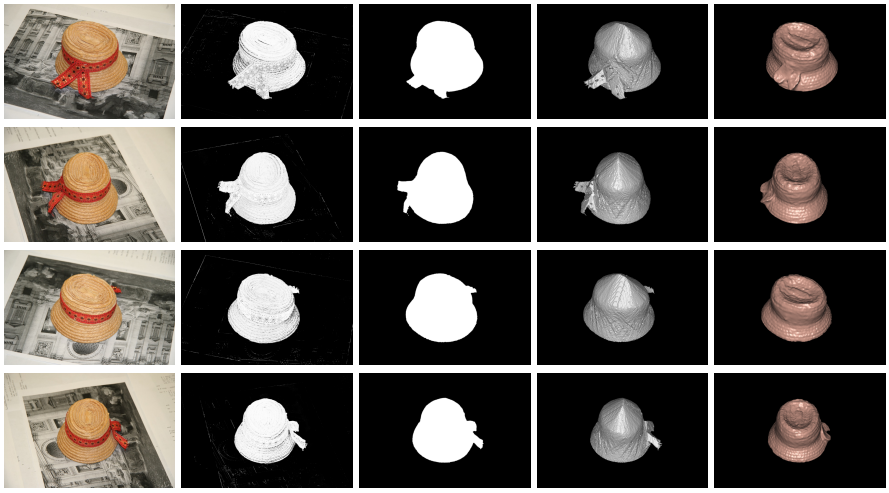
$$\mathrm{E}_{\mathrm{surf}}(\mathcal{O}, \mathcal{B}, \{I_m\}) \quad = \quad \sum_{(v_i, v_j) \in \mathcal{E},\ \substack{v_i \in \mathcal{O} \\ v_j \in \mathcal{B}}} \max_m \ e^{-\beta Z_m(v_i, v_j)} \tag{10}$$

6 Experiments

In our experiments we calibrate a sequence of images. For the acquisition of test sequences we derived our own automatic calibration method. We assume the object lies on a textured plane and thus derive correspondences, using interest points and SIFT descriptors [8], and estimate plane-to-plane homographies. A modified version of the linear calibration from planar homographies in [15] then provides a initial input for non-linear optimisation, via bundle adjustment [4], of the camera intrinsic parameters and individual poses.

For the statue sequence of Figure 4, planar calibration was not possible and therefore the camera intrinsic parameters were calibrated separately followed by the application of structure from motion techniques [4] to track correspondences on the object itself and thus provide an initialisation for bundle adjustment.

The system used to provide the results was based on an Intel Xeon Processor with 4 GB of RAM running at 2.6 GHz. The unoptimised code required 20 minutes per iteration for the hat sequence with a voxel array size of 230^3. The complexity scales linearly with the number of pixels and the majority of the time is spent fitting and evaluating the colour mixture models. We now present the performance of the method on two 'real world' image sequences.



(a) (b) (c) (d) (e)

Figure 3: Fully automatic registration, segmentation and reconstruction of the hat sequence (4 out of 30 images). *The fixation condition is applied to the image sequence (a) and used to initialise the colour models. The colour models converge to a capture all the object colours (b) and the graph-cut at convergence produces the correct silhouettes (c) and visual hull (d) which may be used to produce a reconstruction of the object (e).*

Hat Sequence

Figure 3 shows the segmentation results for a sequence of 30, 4 megapixel images of a hat. The silhouettes and visual hulls shown were obtained after 3 iterations. In addition to the segmentation process being performed without user input, our planar calibration procedure is also fully automatic and thus no user interaction, other than capturing the images themselves, was required to complete this segmentation. Figure 3 (e) shows a 3D model reconstruction of the hat using an implementation of [5].

The object colour model probabilities, Figure 3 (b), are observed to have correctly captured all the object colours resulting in the accurate visual hull of Figure 3 (d) with corresponding silhouettes Figure 3 (c).

Statue Sequence

The method was also tested on a much more challenging image sequence. The statue sequence consists of 69 images of a statue in the British Museum. The images were captured by hand at a resolution of 5 megapixels. As can be seen from the example images of Figure 4 (a), the background continually changes including both the surrounding building and numerous different people and hence a very wide range of colours. The lighting conditions also change dramatically as can be seen by comparing the 3^{rd} and 4^{th} images of Figure 4 (a).

We have compared our results with an implementation of [9] applied independently to each image. Instead of user interaction, we supply the 2D method with the same initialisation as the one supplied in the 3D segmentation algorithm, namely the fixation points. Figure 4 (b) provides the silhouettes and Figure 4 (c) shows the visual hull resulting from the combination of the 2D segmentation results.

The segmentation results after 9 iterations of our method are given in Figure 4 (d). The majority of the silhouettes are registered accurately offering significant improvements over the independent 2D segmentations. We observe that the left leg of the statue is not completely recovered. Since the leg structure is hollow, roughly half of the views see the inside of the sculpture which is different in colour and has failed to be modelled by the priors. Whilst the intersection of 2D binary silhouettes results in removing the whole leg, Figure 4 (c), our 3D segmentation algorithm tries to resolve the discrepancy between the silhouette coherency which results in capturing half the leg as object.

It may also be observed that the regularisation of the reconstruction has underestimated the concavity between the legs. Again this is mostly due to the colour model failing to provide a sharp prior classification in the presence of a cluttered background. Since the gap is only seen in a couple of images the resulting surface falls back on the coherency constraints of the remaining views. Thus the failures of the colour models (for example the same colour in object and background in the statue sequence) are compensated for, to an extent, by the enforcement of spatial coherency. Further improvements could be made my improving the object and background models, for example by including texture information.

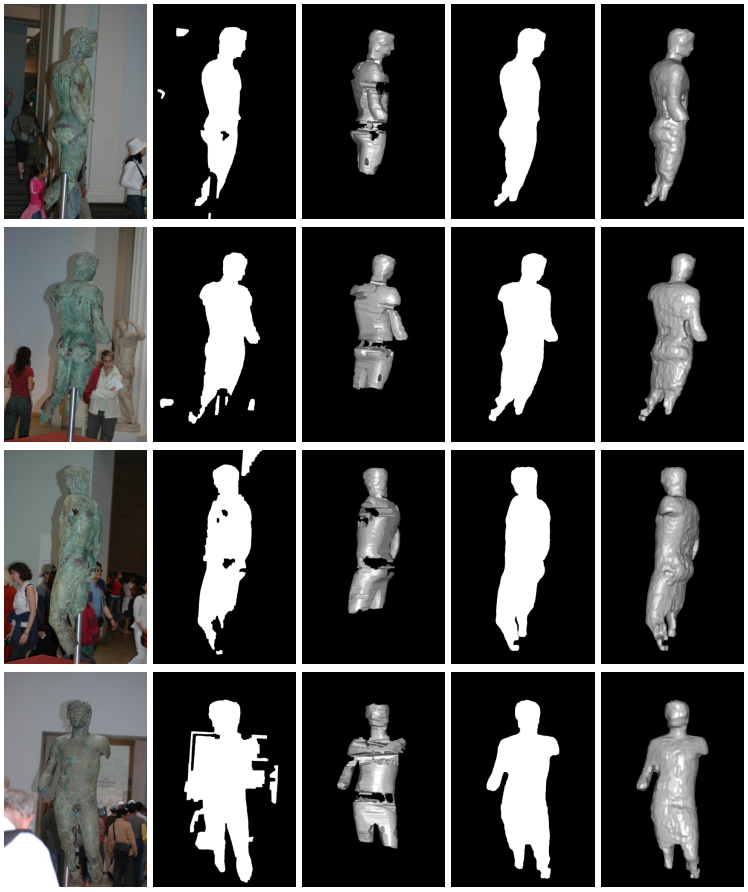
7 Conclusions

The results of our experiments indicate that our method confers considerable advantages for automatic object segmentation over the best 2D algorithms. The volumetric approach makes use of the silhouette coherency constraint to perform the segmentation in 3D, allowing the object to be segmented in all images simultaneously. This allows us to combine the learnt colour model, containing information from all views of the object, with a 3D shape prior to produce a more accurate result. We have also shown that it is possible to exploit a fixation constraint in order to initialise an iterative estimation algorithm to converge to the visual hull of an object observed in multiple views and thus avoid the need for any interaction from the user, thus making the whole process automatic.

Acknowledgements
This work is supported by the Schiff Foundation and Toshiba Research Europe.

References

[1] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
[2] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive GMMRF model. In *Proc. 8^{th} Europ. Conf. on Computer Vision*, pages 428–441, 2004.
[3] Y. Boykov and M. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *Proc. 8^{th} Intl. Conf. on Computer Vision*, pages 105–112, 2001.
[4] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. CUP, 2004.
[5] C. Hernández and F. Schmitt. Silhouette and stereo fusion for 3d object modeling. *Computer Vision and Image Understanding*, 96(3):367–392, December 2004.
[6] C. Hernández, F. Schmitt, and R. Cipolla. Silhouette coherence for camera calibration under circular motion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(2):343–349, 2007.
[7] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(2):147–159, November 2004.
[8] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Intl. J. of Computer Vision*, 60(2):91–110, 2004.



(a) (b) (c) (d) (e)

Figure 4: A single 3D segmentation improves multiple independent 2D segmentations. *The statue sequence contains 69 images of a statue observed in a continually changing background environment (a). Independent 2D segmentation results in incoherent silhouettes (b) and a poor visual hull (c). After convergence, our method produces silhouettes (d) and a visual hull (e) with the correct topology.*

[9] C. Rother, V. Kolmogorov, and A. Blake. "grabcut": interactive foreground extraction using iterated graph cuts. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 309–314, 2004.

[10] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 519–528, 2006.

[11] D. Snow, P. Viola, and R. Zabih. Exact voxel occupancy with graph cuts. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 345–353, 2000.

[12] G. Vogiatzis, P. H. S. Torr, and R. Cipolla. Multi-view stereo via volumetric graph-cuts. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 391–398, 2005.

[13] J. Wang, P. Bhat, R. A. Colburn, M. Agrawala, and M. F. Cohen. Interactive video cutout. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, pages 585–594, 2005.

[14] A. Yezzi and S. Soatto. Stereoscopic segmentation. *Intl. J. of Computer Vision*, 53(1):31–43, 2003.

[15] Z. Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1330–1334, 2000.

Zipf's Law in Image Coding Schemes

Michael Crosier and Lewis D Griffin
Computer Science
University College London
London, WC1E 6BT, UK
m.crosier@cs.ucl.ac.uk

Abstract

Zipf's Law describes a power law between the frequency and rank of an event which is observed, most famously, in word distributions in human languages [1]; where it has been argued [2] that it represents an optimally efficient communication code. From an analogy between image coding schemes and natural language, we hypothesize that conforming to Zipf's Law is a necessary property of a good image description code, i.e. one that provides optimal local descriptors for vision tasks such as object categorization and recognition.

Following the analogy between images and documents, we have developed an 'alphabet' of Basic Image Features (BIFs) [3]. This alphabet allows us to label each pixel, at each scale, with one of a small number of letters. 'Words' in this approach are then fixed square patches of BIF 'letters'.

We test whether the frequency statistics of BIF words obeys Zipf's law, and find that as we vary certain parameters the Zipfness varies systematically. Remarkably, we find that for certain parameter settings we get extremely Zipf behaviour. Moreover the encoding at these settings looks optimal to eyeball measure.

1 Introduction

1.1 Local descriptors

There has recently been a great deal of interest in using local descriptors (e.g. [4], [5]; see [6] for an overview and evaluation) to perform vision tasks such as object categorization [7] and recognition [4].

We are interested in the properties of such local cues which make them suited for these types of tasks. We argue that the graded rarity of descriptors in natural images is an important factor in their effectiveness: If a cue appears in all images, regardless of content, it will be a poor discriminator of objects; as it will if it only ever appears in a particular view of some object instance. Here we explore the frequency statistics of cues from different image coding schemes across natural images.

1.2 Zipf's Law

If counts of word occurrences in a corpus of natural language are sorted into descending order, a power law is observed between the rank r and frequency p of a word so that $p \propto r^{-\alpha}$ with α close to unity. Named after George Zipf, who first noted this relationship for English text [1], Zipf's Law has been empirically observed for a wide range of both human and non-human languages, for example Russian [8] and the whistles of bottlenose dolphins [9].

Motivated by results [2] showing that, under certain assumptions, an information source must obey Zipf's Law in order to ensure optimal efficiency of communication; and by an analogy between image coding schemes and natural language; we hypothesize that conforming to Zipf's Law is a necessary property of a good image description code.

1.3 Coding schemes

Following the analogy between images and documents, we have developed an 'alphabet' of Basic Image Features (BIFs) [3]. This alphabet allows us to label each pixel, at each scale, with one of a small number of letters. At present we have two-, three- and six-letter alphabets based on different sets of Gaussian derivative filters (see figure 1). 'Words' in this approach are fixed square patches of BIF 'letters'.

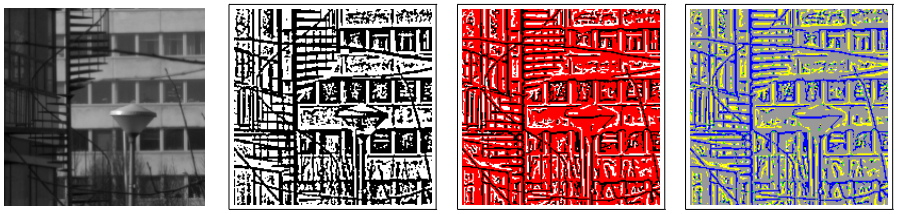


Figure 1: A section of an image from the van Hateren collection [10], encoded respectively with 2, 3 and 6 label BIF schemes.

The BIF coding schemes correspond to a partitioning of filter response space [3]: In the 2-letter scheme, we take the sign of a Laplacean of Gaussian filter (using [11]). For the three letter alphabet, we compare the absolute value of the Laplacean with the gradient magnitude, taking the sign of the Laplacean if $\frac{\sigma^2}{4}\left(I_{xx}+I_{yy}\right)^2 > \left(I_x^2+I_y^2\right)$ where I_x etc. are image derivatives computed at scale σ using Gaussian derivative filters, and a third label otherwise. In both schemes, points are then subsampled at a distance of Δ pixels.

1.4 Testing the 'Zipf Hypothesis'

In this paper we explore the degree to which the frequency statistics of 'words' in natural and synthetic images, encoded with our 2-label ('2-BIFs') and 3-label ('3-BIFs') systems, adhere to Zipf's Law.

We present results which, for certain parameter settings, exhibit strikingly Zipf-like behaviour. Further, these settings seem to correspond well to those which we would choose by eyeball measure. The work presented here will form a basis for testing our 'Zipf Hypothesis' through employing a range of the codes explored here in vision tasks such as object categorization.

2 Methods & Results

2.1 Counting

We counted occurrences of overlapping square patterns ('words'), of sizes 1x1 up to 10x10 pixels, in images which had been encoded with one of our BIF schemes. These counts were then pooled together (so a typical pixel was counted in one 1x1 patch; four 2x2 patches; nine 3x3 patches etc.) and ranked in order of decreasing frequency.

For each size of pattern, approximately 14.6 million patches were drawn evenly from a set of 100 encoded images. For a given area of image, it is possible to extract one more $(d-1)\,\mathrm{x}\,(d-1)$ patch than $d\,\mathrm{x}\,d$ patches in each direction and so we normalised the image area to ensure that the same number of patches were counted of each size.

In order to economise on memory expenditure by reducing the number of patterns recorded, our counting algorithm takes a cutoff value for the least number of occurrences of some pattern across an ensemble of images which we consider to be of interest. This has the effect of cropping the number of ranked patterns in our results, but causes no bias in the distribution of those ranks which are recorded. Indeed, in the absence of such an explicit cutoff its value would default to 1: no pattern will be counted which does not occur at least once in some image. We then proceed iteratively as follows: For each patch size $d\,\mathrm{x}\,d$, we scan through each encoded image, selecting a patch at each position. We extract its four $(d-1)\,\mathrm{x}\,(d-1)$ subpatches and check to see that these have all been recorded sufficiently often (for otherwise it is impossible that the parent patch will occur frequently enough); if so, we increment the count for that patch. Most patterns will occur less often than their smaller subpatches and so, after scanning through all $d\,\mathrm{x}\,d$ patches in all images, we scan through our table of counts and remove references to any pattern which does not occur at least *cutoff* times; before moving on to larger patches.

We consider only patches up to 10x10 pixels for computational reasons. There is a possibility that this could introduce a small bias into our results. However, we argue that since, out of the approximately 300000 patterns in each experiment which occur more frequently than our cutoff, only around 1% of these are of size 10x10; that the number of 11x11 or larger patterns which we are excluding from our results will be vanishingly small.

Four sets of images were considered, one natural and three synthetic:

- Natural images from the van Hateren database [10], containing varied views of countryside, forests, lakes and buildings.
- Phase randomised noise images, based on the power spectra of these same natural images.
- Images generated using a dead leaves model [12]. A large number of opaque discs with randomly sampled positions, radii and luminances are 'stacked' so

that nearer discs occlude those further away. These images capture the occlusive nature as well as approximate scale invariance of natural images.

- 'Coin-flip' random images. Each pixel of these was selected independently from either 2 labels (where the images were to be compared with 2-BIF coded images) or 3 labels (for comparison with 3-BIF coded images). The outcome probabilities for each label were determined by the relative frequencies of that label in the sets of encoded natural images. Thus these images are comparable not with the grey-scale images upon which we employ our coding schemes; but with the 2- or 3-label results of these encodings.

2.2 Zipfness measure

On a log-log plot of frequency vs. rank, Zipf's Law is characterised by a straight line with slope -1. From the one-dimensional space of all such lines we select the best-fitting, and use as our Zipfness measure the weighted sum of squared errors between this and the data, normalised by the log of the number of ranks considered, i.e.

$$\frac{\min_{\beta} \sum_{r=1}^{R} \frac{1}{r} \left(\log f_r - (\beta - \log r)\right)^2}{\log R}, \tag{1}$$

where r is rank, f_r frequency and R the total number of distinct patterns whose occurrences have been counted.

The weighting ensures that equally sized parts of the log-log plot contribute equally to the measure. Normalisation is necessary to compensate for the fact that, for differing parameters, different numbers of patterns appear sufficiently often to be recorded by our counting algorithm.

2.3 Results

We calculated the frequency distributions of patches for each of our four image types coded using either a 2- or 3-label BIF scheme, over a range of parameters.

Because of the well-known scale invariance of natural images (and its replication in our synthesized images) one would expect that for a given ratio between filter scale σ and spatial sampling Δ, varying σ or Δ should make no significant difference to the distribution of frequencies: this is indeed what we find. We therefore define a new parameter $k = \sigma / \Delta$ to be our single degree of freedom in each coding scheme, and study how varying k affects the frequency statistics of BIF words for each BIF alphabet and image type.

Figures 2 and 3 show how frequency distributions vary over k: for small k (i.e. very sparse subsampling), pixels in the encoded images will be essentially independent (see fig. 5, bottom-right) and so the distribution of patches of a given size will be determined almost solely by the relative frequencies of each label. For our 2-label

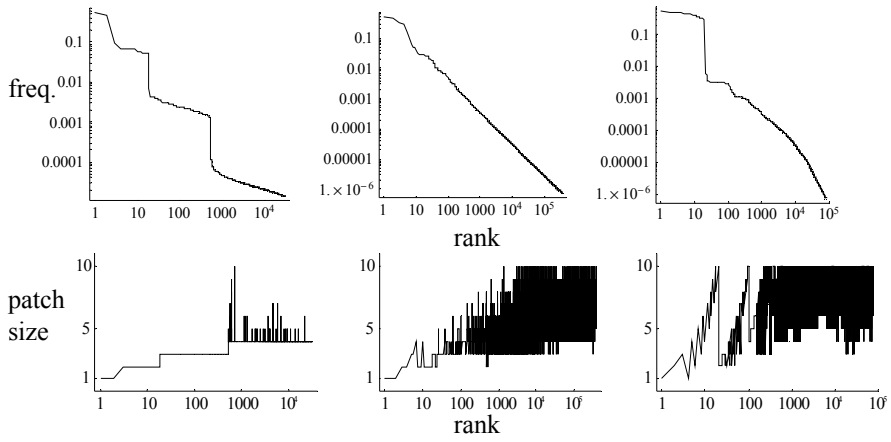


Figure 2. Top: Log-log plots of frequency of occurrence against rank for natural images encoded with 2-BIFs. From left to right, the ratio k between filter scale and spatial sampling takes the values 0.2, 1.5 and 12. Bottom: The size of patches at each rank in the same results. Similar frequency distributions are observed for phase randomised noise and dead leaves model images, although patch size distributions vary slightly: in particular, for intermediate values of k the results for dead leaves images tend to show peaks in patch size for low ranking patterns, caused by large areas of zero contrast in some of the original images. This is explored further in figure 8.

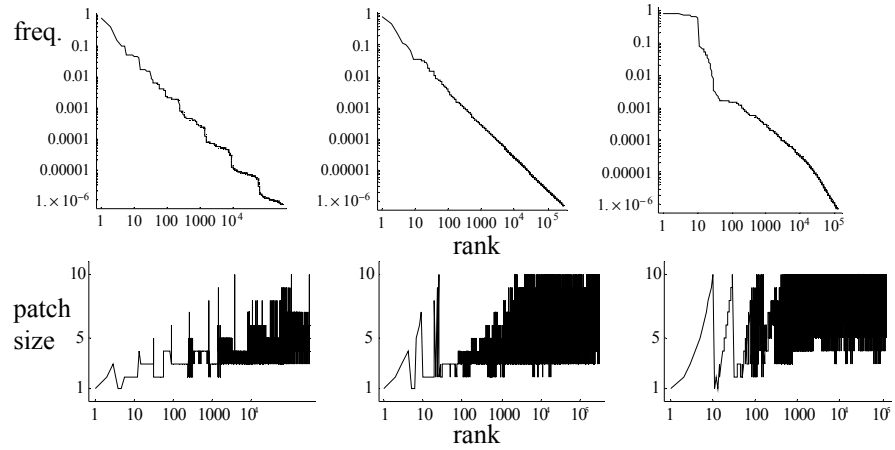


Figure 3:As figure 2, but for natural images encoded with 3-BIFs. From left to right, k takes the values 0.2, 1.0 and 12. The main difference from figure 2 is in the results for small k (i.e. undersampled coding) where the rank-frequency plot is less step-like. This is a consequence of the relative frequencies of the labels in our 3 label system being less symmetric than those for 2-BIFs, resulting (as shown in figure 5) in large areas of the same (red) label in the encoded images.

system, each label occurs with approximately the same frequency and so patches of each size are close to being uniformly distributed, resulting in the stepped plot seen in figure 2. In figure 3 (3-BIFs), this effect is less pronounced due to the uneven label frequencies in 3-BIF encoded images; but is still present.

For large k, on the other hand, encoded images will be characterised by large connected areas of the same label (see fig. 5, bottom-left). Most patches observed, of all sizes, will therefore be single-label patterns, and any moderately complex pattern will be very rare (see fig. 6). This accounts for the high initial frequency and consequent steep dropoff seen in figures 2 and 3.

Whilst encodings for small k are undersampled; and for large k oversampled; those for a reasonably broad mid-range of k-values appear 'just right': as in the top-right of figure 5. Remarkably, this seemingly optimal mid-range corresponds quite well to those distributions which closely follow Zipf's Law, as in the middle parts of figures 2 and 3.

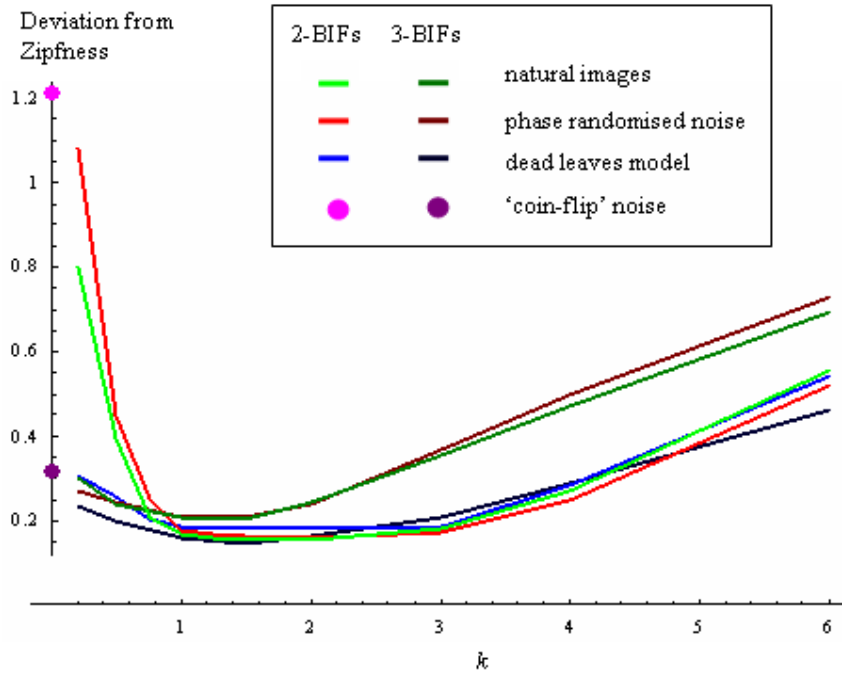


Figure 4: Variation of 'Zipf-dissimilarity' score with changing k for all image types and coding schemes. The 'coin-flip' images were not encoded in the same way as the other image types, and so do not vary over a parameter k. They are shown here as having $k = 0$ since they represent a logical limiting value for the other encoding schemes as $k \to 0$: As k decreases, encoded pixels are sampled increasingly sparsely and so become more independent of their neighbours. 'Coin-flip' noise models an encoded image in which the pixels are all independent.

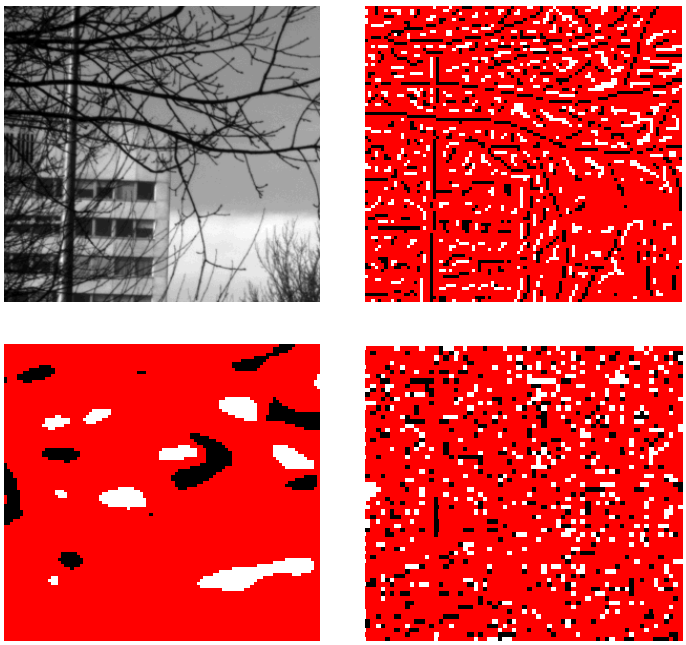


Figure 5: A section of a natural image from the van Hateren database, and 3-BIF encodings of the same image part with different values of k (anticlockwise from bottom-left: k =6.0, 0.2, 1.0. Recall that $k = \sigma/\Delta$, the ratio between filter scale and spatial sampling: In these examples, Δ was kept fixed and σ varied to produce the stated values of k while allowing the section of image to be encoded into the same number of pixels). The top-right encoding ($k = 1.0$) clearly displays more of the original image's structure. Comparison with the results in figure 4 shows that codings with the lowest Zipf-dissimilarity scores also appear optimal to eyeball measure.

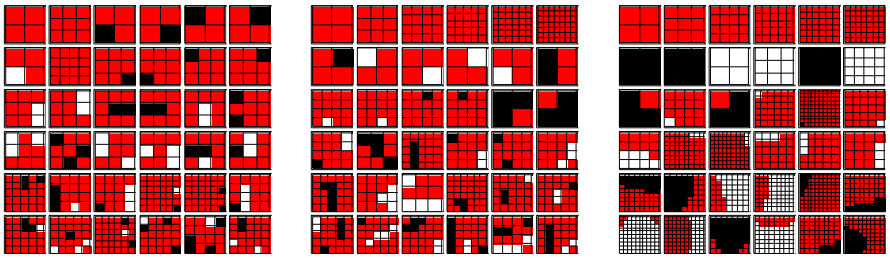
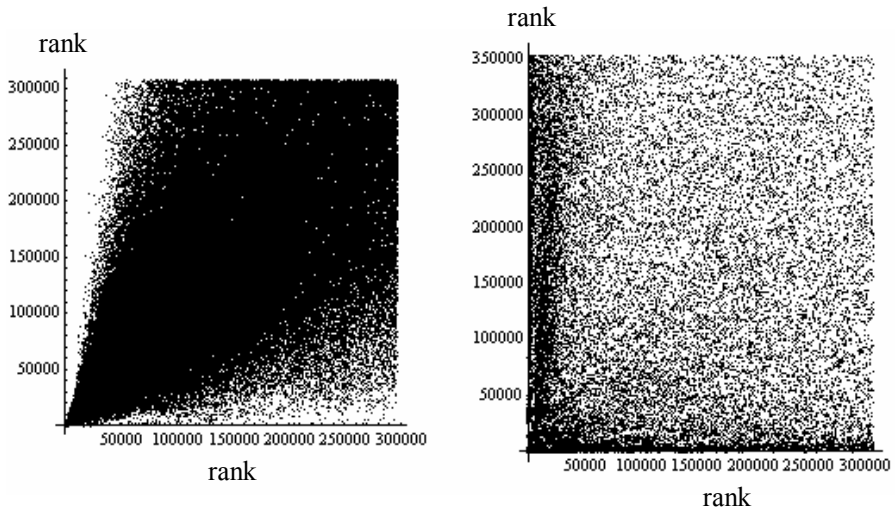


Figure 6: Examples of 3-BIF 'words' occurring in natural images for different values of k (left to right: k =0.2, 1.0, 12). The top row of each set shows the 6 most popular patches; the second row the 11th to 16th most frequent; the third row the 101th to 106th most frequent; and so on to the 6 patches ranked around 100000 in frequency of occurrence.



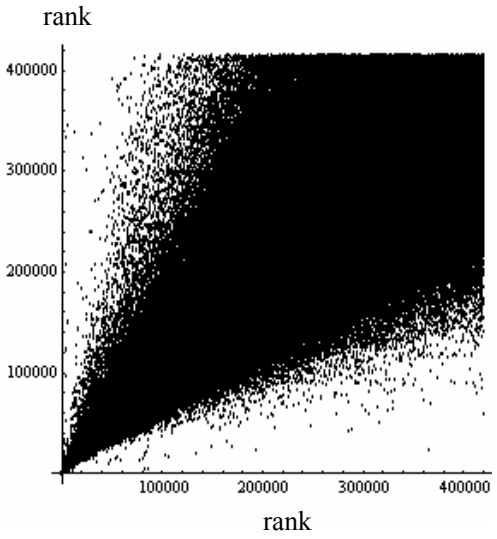


Figure 7: Stability of pattern rank. Top left: The rank of patterns occurring in both natural and phase randomised noise images encoded with 2-BIFs with $k = 1.0$ (correlation coefficient ρ =0.68). Top right: The rank of patterns occurring in natural images encoded with 2-BIFs, first with $k = 1.0$ and then $k = 4.0$ (correlation coefficient ρ =0.06). Right: The rank of patterns occurring in two different sets of natural images, both encoded with 2-BIFs with $k = 2.0$ (correlation coefficient ρ =0.84). Far greater rank-stability is seen between different types of images encoded in the same way than between the same images encoded with different ratios (k) between filter scale and spatial sampling.

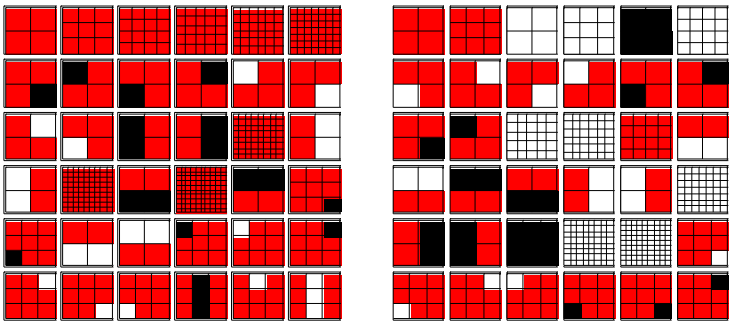


Figure 8: As fig. 6, but showing on the left patterns occurring in natural images encoded with 3-BIFs and $k = 1.0$; and on the right patterns occurring in dead leaves model images encoded with 3-BIFs and $k = 1.0$. The results for phase randomised noise images are very similar to those for natural images. The large connected areas of white or black labels are caused by 3-BIF-encoding relatively large areas of zero-contrast (which are very seldom seen in natural or phase randomised images) in the dead leaves images. This differing behaviour is mirrored in figure 4, where the results for 3-BIF encoded dead leaves model images appear more Zipf-like than results for natural images or phase randomised noise

This 'mid-range' is made more explicit in figure 4, which shows the full range of Zipfness scores for all the image types and encoding schemes studied. For both 2- and 3-label systems, there is very good correlation between the scores for natural and phase randomised noise images, and as expected (see caption of fig. 4) both of these seem to be tending towards the results for 'coin-flip' noise as $k \to 0$. Images encoded with 2-BIFs appear to achieve optimal Zipfness around $k = 1.5$; and those encoded with 3-BIFs around $k = 1.0$.

Figure 4 shows that the frequencies of occurrence of patterns in natural and phase randomised noise images (and, for 2-BIF coding, dead leaves model images) encoded in the same way are distributed in a similar fashion. Figure 7 expands on this by showing (left-hand side) that the same patterns occur at reasonably similar ranks in these distributions. By contrast (fig. 7, right-hand side), when the same images are encoded with different parameters there is almost no correlation between the ranks of individual patterns in the two codes.

3 Conclusions and Further Work

We have tested whether the frequency statistics of BIF words obeys Zipf's law. We find that as we vary the ratio between filter scale and spatial sampling the Zipfness varies systematically. We have found that for certain parameter settings we get extremely Zipf behaviour. Moreover the encoding at these settings looks optimal to eyeball measure.

We are currently testing whether Zipfness is a predictor of performance of the code when it is used for the task of texture classification.

Acknowledgements

EPSRC-funded project ‘Basic Image Features’ EP/D030978/1.

References

[1] G. K. Zipf, *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*: Addison-Wesley, 1949.

[2] B. Mandelbrot, "Contribution a la the´orie mathe´matique des jeux de communication.," in *Publications de l’institut de statistique de l’universite´ de Paris*, vol. 2, 1952, pp. 1–124.

[3] L. D. Griffin and M. Lillholm, "Feature category systems for 2nd order local image structure induced by natural image statistics and otherwise.," presented at SPIE 6492(09):1-11, 2007.

[4] D. G. Lowe, "Object recognition from local scale-invariant features," presented at Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on, 1999.

[5] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Gray Scale and Rotation Invariant Texture Classification with Local Binary Patterns," in *Computer Vision - ECCV 2000*, vol. Volume 1842/2000, *Lecture Notes in Computer Science*: Springer Berlin / Heidelberg, 2000, pp. 404-420.

[6] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, pp. 1615-1630, 2005.

[7] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," presented at Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on, 2003.

[8] A. Gelbukh and G. Sidorov, "Zipf and Heaps Laws' Coefficients Depend on Language," in *Computational Linguistics and Intelligent Text Processing : Second International Conference, CICLing 2001, Mexico-City, Mexico, February 18-24, 2001, Proceedings:*, 2001, pp. 332.

[9] B. McCowan, S. F. Hanser, and L. R. Doyle, "Quantitative tools for comparing animal communication systems: information theory applied to bottlenose dolphin whistle repertoires," *Animal Behaviour*, vol. 57, pp. 409-419, 1999.

[10] J. H. van Hateren and A. van der Schaaf, "Independent component filters of natural images compared with simple cells in primary visual cortex," *Proceedings of the Royal Society B: Biological Sciences*, vol. 265, pp. 359-366, 1998.

[11] B. M. ter Haar Romeny, *Front-End Vision and Multi-Scale Image Analysis*: Kluwer Academic Publishers, 2003.

[12] A. B. Lee, D. Mumford, and J. Huang, "Occlusion models for natural images: A statistical study of a scale-invariant dead leaves model," *International Journal of Computer Vision*, vol. 41, pp. 35-59, 2001.

Segmenting Highly Textured Nonstationary Background

David Russell and Shaogang Gong
Department of Computer Science
Queen Mary, University of London
London E1 4NS, UK
{dave,sgg}@dcs.qmul.ac.uk
http://www.dcs.qmul.ac.uk/researchgp/vision

Abstract

Detection of unusual objects amongst a highly textured background is a difficult problem, especially when the texture is manifest in the temporal dimension as well. Outdoor scenes involving waving trees or moving water are examples of such a scenario, but are nevertheless frequently encountered in real world vision applications. By defining a simple but rotationally sensitive Local Binary Pattern (LBP) operator and applying it in a probabilistic sense we present a compact but useful feature for tackling moving textures. But as we demonstrate, this alone is not sufficient for good segmentation in difficult circumstances. Cooccurrence of different features in a pixel's local neighbourhood provides a powerful mechanism for boosting the reliability of the foreground/background decision task. By using the conditional probabilities yielded by pairwise cooccurrence of 4-connected pixels, and casting the problem as one of Combinatorial Optimization, our results show that useful segmentation *is* possible from challenging dynamic backgrounds.

1 Introduction

Effective background modelling is a crucial first stage in most computer vision applications, especially in outdoor environments. The reliability with which potential foreground objects can be identified directly impacts on the efficiency and performance level achievable by subsequent processing stages such as tracking, recognition and threat evaluation. The nature of such a background is intrinsically statistical. Whilst the concept of statistical scene modelling suggests that there is no exact distinction between what constitutes foreground and background, a useful practical definition for surveillance in a busy urban scene is that people and the objects they cause to move are foreground. Buildings, fixtures, trees and permanent objects, together with any environmental change in lighting such as shadow caused by moving clouds, form the background. Critically, we consider that background is in general necessarily amongst foreground, i.e. it can be literally *behind* and *in front* of foreground objects, especially in urban outdoor scenes, as later examples show. The task of the background model in such a setting is to discriminate between the two classes under a potentially wide variety of lighting conditions. Evidently,

confusion might still arise, since trees sway in the wind, tending to become foreground, whilst people park their cars, which are eventually subsumed by the background. The most commonly encountered models are based on per pixel techniques such as adaptive Gaussian Mixture Models [13, 14], or subspace analysis based methods [10, 8], and both approaches have been used with success in many applications. However, all these techniques require and assume that the background in a scene settles quickly into a stationary state, so that the distribution of background pixels becomes stable and tight, although not necessarily continuous. This is often not the case, especially in outdoor scenes.

The focus of this paper is to tackle the more challenging problem of modelling highly textured nonstationary backgrounds, and in particular, segmenting people moving amongst dense nonstationary trees and foliage excited by the wind. Traditionally this has been a difficult problem to solve effectively due to the highly chaotic nature of such image areas containing branches and leaves. The high information content, or *entropy*, of patterns encountered and their temporal behaviour make them inherently incompressible and thus hard to model compactly. With subspace techniques as employed in [10, 8], the eigenvectors of image covariance represent linkage of pixel variations across the entire scene, and are thus inefficient at capturing such independent local stochastic processes. Connectivity in the temporal dimension as exploited by Linear Prediction [14] is also likely to be ineffective due to the lack of cyclic components of intensity at a pixel.

On the other hand, Gaussian Mixture Models (GMM) [13] have been shown highly effective when it comes to acquiring and adapting to the statistical characteristics of behaviour at a pixel. However, high variance (or covariance for a colour image) inevitably implies low selectivity for a Gaussian component, so unless the spread of common pixel values is confined to several narrow modes, there is the danger that a foreground object will fail to be detected reliably. In addition, the GMM is at the opposite extreme from the subspace model when it comes to connectivity: in general it offers no mechanism for support regarding foreground/background decisions between pixels, either local or global.

A further additional requirement for a background model intended for outdoor use is that it must not be severely affected by changes in scene illumination with regard to both intensity and chromaticity, although in some implementations [6] constancy of the latter is used to mitigate the effect of false positives caused by shadows.

From these observations it becomes apparent that a candidate solution should satisfy the following: (1) have a probabilistic basis, (2) encode local pixel patterns, (3) exhibit resilience to lighting variations, (4) provide local support among pixels, and (5) be efficient in implementation. To this end, we propose a solution which embraces three important aspects. Firstly, a rotationally variant simplification of the LBP_8 operator as the image feature reduces susceptibility to illumination changes and provides an initial level of pattern sensitivity. Secondly, a cooccurrence map representing mutual conditional probabilities between adjacent pairs of pixels lends local support to the foreground/background segmentation decisions, encoding a further degree of pattern dependence. Finally, the array of image pixels is treated similarly to a Markov Random Field (MRF), and an optimal realization of the segmentation in terms of pixel labelling in a combinatorial sense is arrived at by a minimum cut on a related graph. Experiments on a challenging dataset involving objects heavily obscured by tree branches demonstrate the advantage of this approach.

2 High Entropy Scenes

Many typical scenes contain areas of high inherent complexity such as specular reflection from disturbed water and chaotic occlusion and appearance variation of vegetation moving under the influence of air flow. From the standpoint of information theory these represent *high entropy* sources [7], whilst signal processing tends to consider the effect spectrally, and refers to sources emitting *wideband noise*. In single frames, the chaos is a spatial property, whilst in video such stochastic variation occurs temporally as well. Exact modelling of the precise characteristics of intensity over time in a high entropy image area is by definition almost impossible: the information is highly incompressible.

From a foreground/background detection point of view we wish to highlight unusual state or behaviour of the objects in view, which might for example entail a person walking in front of a tree in leaf, or perhaps passing *behind* it, causing partial occlusion due to the *background* now being in front of the person of interest. In both cases, the requirement is to identify some less common pixel intensity configurations amongst a potentially broad range. Occlusion of the foreground, as in the second case, merely compounds the detection problem by fragmenting the available useful evidence.

2.1 Rotationally Specific LBP_4

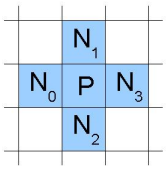
High entropy image content is commonly modelled as texture [15]. This general approach does not encode *exact* pixel configurations, rather *typical* patterns exemplary of the region. The LBP_8 operator described in [9] cleverly encodes a summary of patterns in a 3×3 pixel block into one of ten different codewords in a way which renders it insensitive to both absolute illumination and pattern rotation. These are both crucial attributes in texture analysis. Using such a scheme, segmentation on the basis of texture may be achieved by identifying regions with a similar probability distribution over the ten possible codewords.

But the requirement for foreground/background segmentation is different. We are not interested in regional statistics, but pattern statistics at a pixel, and furthermore, rotational invariance is not only unnecessary, but a hindrance with regard to our modelling requirement. Thus we introduce the concept of a Rotationally Specific Local Binary Pattern (RSLBP) operator for grayscale images, obtained by simplifying LBP_8. As shown in Figure 1 the value of the RSLBP_4 operator at a pixel is given by subtracting the intensity value of the centre pixel from each of its 4-connected neighbours. The sign of the result of each subtraction contributes a single bit to form a 4 bit codeword. For us, the spatial mapping from neighbour to bit position is immaterial as long as it is applied consistently. This rotationally specific texture feature is quick and simple to compute, and yields a compact characterization of two-dimensional image gradient at a pixel fit for our purpose.

Application of the RSLBP_4 operator to an image produces a symbol $S_r = \{0 \ldots 15\}$ at pixel location r. By considering the 16 bin histogram of these symbols at each pixel over a training set of K frames, we obtain an estimate of Probability Density Function (PDF) representing pixel configuration over this feature:

$$p(r = S_r | x, y) = \frac{1}{K} \sum_{k=1}^{K} u \quad \text{where} \quad u = \begin{cases} 1 & \text{if } R(I^T_{x,y,k}) = S_r \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

A query image I^Q may be tested against this simply by evaluating the RSLBP_4 operator at every pixel and obtaining the appropriate probability from the histogram, which in turn is tested against a threshold to yield a rudimentary foreground/background segmentation.



$$RSLBP_4 = \sum_{i=0}^{3} (N_i > P).2^i$$

Figure 1: Kernel for the new RSLBP$_4$ operator: a 4 bit word is composed from the boolean results of thresholding the 4-connected neighbours against the centre pixel.

2.2 Cooccurrence Matrix

In order to provide local support between pixels, we also use the training data to build a cooccurrence matrix between every adjacent pair of 4-connected pixels both horizontally and vertically in the image. This two-dimensional histogram represents the joint probability of two separate RSLBP$_4$ symbols occurring simultaneously at the two adjacent locations. Although conceptually, cooccurrence between pixels horizontally and vertically is the same, from an implementation point of view it is preferable to consider it in two separate arrays, C_h of size $(M-1)\times N\times 16\times 16$ elements, and C_v of size $M\times(N-1)\times 16\times 16$ elements.

$$C_h(x,y,i,j) = \frac{1}{K}\sum_{k=1}^{K} u \quad \text{where} \quad u = \begin{cases} 1 & \text{if } R(I^T_{x,y,k}) = i \;\&\; R(I^T_{x+1,y,k}) = j \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$C_v(x,y,i,j) = \frac{1}{K}\sum_{k=1}^{K} u \quad \text{where} \quad u = \begin{cases} 1 & \text{if } R(I^T_{x,y,k}) = i \;\&\; R(I^T_{x,y+1,k}) = j \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

at location (x,y) where $i,j = \{1,2\ldots 16\}$, $R(\cdot)$ is the RSLBP$_4$ operator, and I^T_k $k = \{1,2\ldots K\}$ is the training set. The cooccurrence matrices at each pixel are normalized to the number of training samples K such that they correctly reflect the joint PDF.

Now consider two horizontally adjacent pixels r and s in a query image I^Q having RSLBP$_4$ symbols S_r and S_s respectively. If on the basis of our information solely about pixel r from the training data we decide that it is background, then we can obtain from the cooccurrence relationship a conditional probability of symbols over pixel s from $C_h(x_r,y_r,S_r,S_s)$. But in order for this to be a valid probability, we have to normalize C_h over its last dimension such that the conditional probability of s given r is:

$$p(s = S_s | r = S_r, x_r, y_r) = \frac{C_h(x_r,y_r,S_r,S_s)}{\sum_j C_h(x_r,y_r,S_r,j)} \tag{4}$$

However, the relationship between r and s is symmetrical, so if s were known to be background then the conditional probability over r comes from a similar expression. We note however, that the normalization constant in the denominator must be obtained by summing along the *third* dimension of C_h this time:

$$p(r = S_r | s = S_s, x_r, y_r) = \frac{C_h(x_r,y_r,S_r,S_s)}{\sum_i C_h(x_r,y_r,i,S_s)} \tag{5}$$

It becomes apparent that these mutually dependent results cannot be acted on sequentially, especially when it is remembered that a pixel is potentially supported by four neighbours. We consider that for any given query image there will be a combination of foreground/background decisions amongst the pixels, i.e. a segmentation by pixel labelling, such that the labelling process is made optimal according to our localized support measure introduced above. To find the optimal labelling of all pixels in a scene, we assert that the problem is an exercise in Combinatorial Optimization and look to a graph cut technique in order to solve it.

3 Combinatorial Optimization

The problem of choosing a label for each pixel in an image from a finite set of labels according to a set of penalty expressions is the essence of discrete optimization. The objective is to separate the pixels according to their labels in the configuration which incurs the least penalty. If the penalty criteria are correctly designed, the optimal separation is useful in some way.

The labelling of pixels from a discrete set is directly equivalent to making a cut on a graph consisting of vertices and edges as shown in Figure 2. In such a graph there is a vertex for each pixel, and a special *terminal* vertex representing each element of the label set. Every pixel node is coupled by an edge to every terminal, but edges also exist between the pixels to represent their interdependencies. According to a scheme of penalties, every edge is assigned a weight determined by the cost of cutting that edge. The optimal solution is obtained by cutting enough edges to leave every pixel connected to exactly one terminal, thereby taking on that terminal's label, and yielding the combination of pixel to terminal assignments which gives the minimum cost cut of the graph and hence the overall problem solution.

It was shown in [4] that for the special case of two labels, an optimal solution can be obtained in polynomial time using the Minimum Cut/ Maximum Flow (MinCut/MaxFlow) algorithm. Fortunately our foreground/background segmentation is just such a binary problem. Segmentation into more regions than this is potentially interesting, but the multi-way cut has been shown to be NP-complete [3], although [2] describes a way of achieving a *local* energy minimum within a constant factor of the *global* minimum by their alpha expansion algorithm.

The graph cut problem has much in common with the solution of Bayesian networks and Markov Random Fields (MRF) [5], whereby a realization of the field encompasses the interdependencies of the nodes. A method described in [12] demonstrates how local support can be achieved by considering the grid of pixels as an MRF utilising the Potts interaction model [11], in which the penalty for separating pixels is a constant. This leads to their goal of overall smoothness in the segmentation, which whilst might look appealing may eventually not be accurate.

In solution of the binary label case by the MinCut/MaxFlow algorithm, one can imagine trying to transport as much water from the *source* node to the *sink* node by a system of pipes having capacity limits equal to the edge weights. When no more capacity can be added to the network, the path traced by the saturated pipes (edges) defines the minimum cut. In our case, the capacity of a pipe depends on which way the water is flowing, i.e. which of its end nodes is joined to the source and which to the sink. This is crucial in

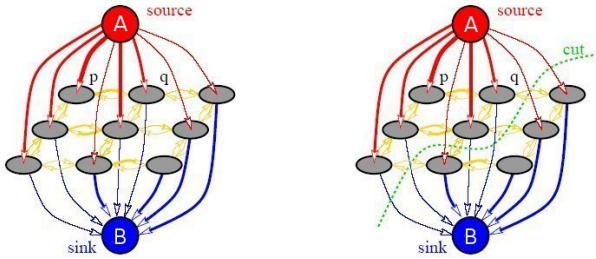
determining which conditional probability, and hence penalty, is applied at the final segmentation. In our algorithm, illustrated for clarity here by only three pixels in Figure 3(a), the cost of a given labelling $\mathcal{L}$ is the energy function

$$E(\mathcal{L}) = \sum_{r \in I^{\mathcal{Q}}} D_r(l) + \sum_{\{r,s\} \in \mathcal{N}} V_{rs}(p(r|s), p(s|r)) \tag{6}$$

consisting of penalty terms D derived from a pixel's probability in isolation of being background and an interaction term V based on conditional probability from cooccurrence. Here $\mathcal{N}$ represents the 4-connected neighbourhood of connections as shown in Figure 2 (not to be confused with the 4-connectivity earlier in RSLBP_4, even though it involves the same pixels). The edge weights illustrated in Figure 3(a) are assigned as follows:

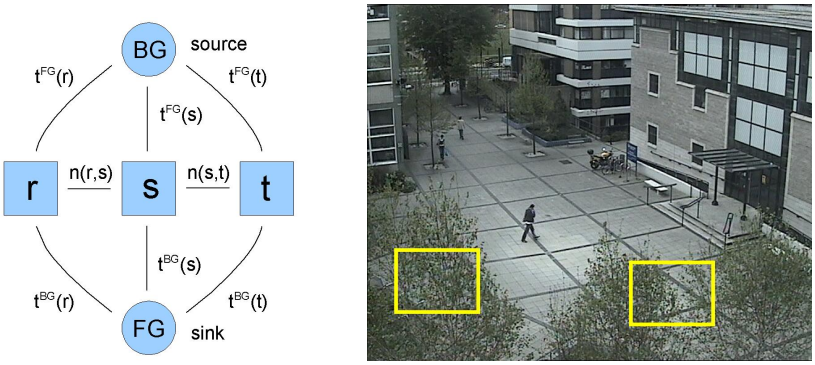
Edge	Forward Capacity	Reverse Capacity
$t^{BG}(r)$	1	1
$t^{FG}(r)$	$\frac{\beta}{(p(r=S_r)+0.01)}$	$\frac{\beta}{(p(r=S_r)+0.01)}$
$n(r,s)$	$\lambda\, p(s=S_s \mid r=S_r, x, y)$	$\lambda\, p(r=S_r \mid s=S_s, x, y)$

The V terms can be seen as a penalty for separating pixels which, according to cooccurrence, should belong together and to the background. To cause them to end up separated, one would have to have a very low individual probability of occurring. The constants β and λ control the magnitude of the effect of the D and V penalties relative to each other, and also to the unity penalty assigned to the cost of being background.



(a) A graph $\mathcal{G}$ (b) A cut on $\mathcal{G}$

Figure 2: Graph for an array of only 9 pixels: *source* and *sink* nodes represent the two classes A and B. A cut must separate A and B: the MinCut/MaxFlow algorithm finds the cheapest. A practical graph contains a node for *every* image pixel. Figure taken from [1].



(a) (b)

Figure 3: (a) More detailed graph for an array of only 3 pixels, showing Background as the *source* label and Foreground as the *sink*. Terminal and neighbourhood link edge weights are shown as t and n respectively. Cutting a lower t-link joins a pixel to the BG label incurring cost t^{BG} (b) Scenes chosen for the experiment lie within highlighted windows.

4 Experiment

To demonstrate the effectiveness of our algorithm using RSLBP_4 and MinCut-MaxFlow, the challenging scene shown in Figure 3(b), containing a leafy tree in a courtyard, was chosen. The leaves move significantly in the wind whilst people pass *behind* the tree, but remain visible through the foliage. From a dataset of 2500 monochrome frames of size 128×96 pixels, 2000 are used as the training data to build the probability distributions and the cooccurrence matrices C_h and C_v. From the remaining frames we select an interesting subset in which a person enters the scene from the top right and walks towards the camera. We compare our RSLBP_4 operator with the standard LBP_8 operator, and also with a more primitive feature: a 16 level grayscale derived by merely truncating the pixel intensity to 4 bits. The MinCut algorithm and the previously tabulated weighting scheme was applied in all cases, and results are shown in Figure 4. We further demonstrate contribution of the MinCut stage with a comparative result in which it is *not* used: Figure 5 shows what happens when individual pixel probabilities alone are used for segmentation using the RSLBP_4 operator. Even when the foreground detection threshold is optimized manually to 0.045, there is only a hint of the presence of a person, and most of the foreground pixels are noise. Figure 6 provides further evidence in support of the RSLBP_4 and MinCut combination, with images from the right hand window in the scene of Figure 3(b).

Although the Combinatorial Optimization algorithm chooses discrete labels as its solution, the notion of a detection threshold still exists in the form of the relative scaling of the various edge weights. In our implementation, β controls the effect of the pixels' individual probabilities, whilst λ regulates the influence of the inter-pixel support. In each case, since the probabilities vary between 0 and 1, the two constants act as maximum values for their own particular type of edge. Choosing $\beta = 7$ and $\lambda = 10$ scales the optimization favourably when the t^{BG} edges are set to unity.

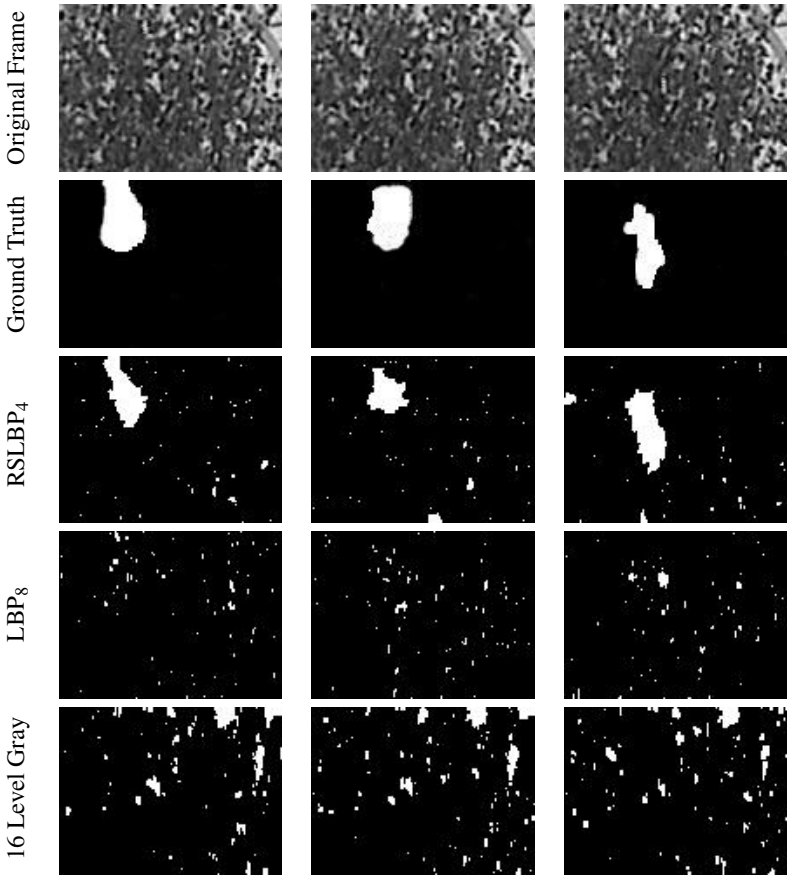


Figure 4: Three frames from the left hand window of Figure 3(b) in which a person walks behind a tree. From top to bottom: Original, Ground Truth, using $RSLBP_4$ operator, using LBP_8 operator, and using 16 level grayscale, all *with* MinCut. Note that our $RSLBP_4$ operator is the only one to produce a useful segmentation here.

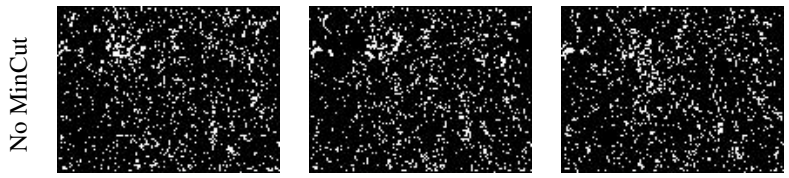


Figure 5: The same three frames using $RSLBP_4$ but *without* MinCut, and hence no local support. The person is barely discernible amongst the noise. LBP_8 and Grayscale are similarly ineffectual without the vital MinCut stage.

Our $RSLBP_4$ operator can generate 16 different values as currently defined, leading to a cooccurrence matrix with only 16×16 entries. This compactness is convenient for two practical reasons. Firstly the memory required to store the inter pixel data is manageable, and secondly the quantity of training data to adequately estimate it remains modest. LBP_8 generates only 10 possible values, but as the experiments show, its rotational invariance renders it useless for our purpose. A rotationally *variant* version of LBP_8 generates 59 combinations and is thus, according to the previous arguments, not so attractive.

Overall the favourable segmentation afforded by $RSLBP_4$ in our results in Figures 4 and 6 strongly supports the idea that it is a better choice than the other two commonly encountered features for the current application. Furthermore, the comparison between Figures 4 and 5 show clearly that the graph cut technique contributes enormously to the quality of the segmentation. We believe that the 'double level' of local spatial support afforded by the partnership of the two techniques is the reason for the distinctive result.

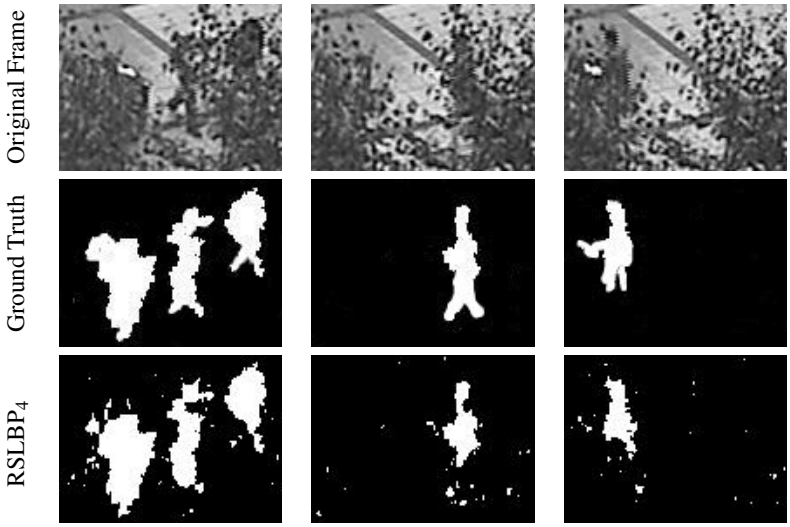


Figure 6: Further results using $RSLBP_4$ and MinCut from the right hand window in Figure 3(b) in which people pass behind trees. The algorithm succeeds in identifying unusual objects in spite of considerable local clutter from the leaves.

5 Conclusion

We have introduced a simple new operator $RSLBP_4$ based on existing LBP methods, and have shown how it can be applied to advantage in the foreground/background segmentation of highly textured dynamic scenes. We claim that its sensitivity to rotation, but resilience to overall illumination variations, both contribute vitally to its success in this application. The restricted range of output symbols of $RSLBP_4$ permits tractable acquisition of adjacent pixel cooccurrence data. We have shown that such data may be used to

construct a graph, of which the minimum cost cut facilitates mutually supporting inferences between pixels, leading to a useful segmentation which would not have been easy to arrive at otherwise. Although a model involving separate collection of training data is described here, it is anticipated that an adaptive online derivative would also be possible, and that this would provide a useful direction for further research.

6 Acknowledgment

The authors would like to thank Vladimir Kolmogorov for use of his C++ implementation of the MinCut/MaxFlow algorithm which is available at:
http://www.adastral.ucl.ac.uk/ vladkolm/software.html.

References

[1] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in computer vision. *IEEE PAMI*, 26(9):1124–1137, 2004.

[2] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE PAMI*, 23(11):1222–1239, 2001.

[3] Elias Dahlhaus, David S. Johnson, Christos H. Papadimitriou, P. D. Seymour, and Mihalis Yannakakis. The complexity of multiterminal cuts. *SIAM J. Comput.*, 23(4):864–894, 1994.

[4] L. Ford and D. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.

[5] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE PAMI*, 6(6):721–741, November 1984.

[6] T. Horprasert, D. Harwood, and L.S. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *ICCV Frame-Rate WS*, Kerkyra, Greece, 1999.

[7] H. Hung and S. Gong. Quantifying temporal saliency. In *British Machine Vision Conference*, pages 742–749, September 2004.

[8] Y. Li. On incremental and robust subspace learning. *PR*, 37(7):1509–1518, 2004.

[9] T. Ojala, M. Pietikainen, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *PR*, 29(1):51–59, January 1996.

[10] N. Oliver, B. Rosario, and A. Pentland. A Bayesian computer vision system for modelling human interactions. *IEEE PAMI*, 22(8):831–843, August 2000.

[11] Renfrey B. Potts. Some generalized order-disorder transformation. In *Transformations, Proceedings of the Cambridge Philosophical Society*, volume 48, pages 106–109, 1952.

[12] Konrad Schindler and Hanzi Wang. Smooth foreground-background segmentation for video processing. In *ACCV (2)*, pages 581–590, 2006.

[13] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE CVPR*, pages 246–252, Colorado, 1999.

[14] K. Toyama, J. Krumm, B. Brummit, and B. Meyers. Wallflower: Principles and practice of background maintenance. In *IEEE ICCV*, volume 1, pages 255–261, Kerkyra, Greece, 1999.

[15] C. S. Zhu, N. Y. Wu, and D. Mumford. Minimax entropy principle and its application to texture modeling. *Neural Computation*, 9(8), November 1997.

Improving Spatial Support for Objects via Multiple Segmentations

Tomasz Malisiewicz and Alexei A. Efros
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Sliding window scanning is the dominant paradigm in object recognition research today. But while much success has been reported in detecting several rectangular-shaped object classes (i.e. faces, cars, pedestrians), results have been much less impressive for more general types of objects. Several researchers have advocated the use of image segmentation as a way to get a better spatial support for objects. In this paper, our aim is to address this issue by studying the following two questions: 1) how important is good spatial support for recognition? 2) can segmentation provide better spatial support for objects? To answer the first, we compare recognition performance using ground-truth segmentation vs. bounding boxes. To answer the second, we use the multiple segmentation approach to evaluate how close can real segments approach the ground-truth for real objects, and at what cost. Our results demonstrate the importance of finding the right spatial support for objects, and the feasibility of doing so without excessive computational burden.

1 Introduction

In the early days of computer vision, image segmentation fit neatly into the well-defined object recognition pipeline. First, in the image processing stage, low-level features (edgelets, corners, junctions) are detected. Next, in the segmentation stage, these features are used to partition the image into regions. Finally, in the recognition stage, the regions are labeled with object identities (and possibly even 3D object models!). Alas, it was soon evident that hopes of computer vision being solved in such a simple and elegant way were overly optimistic. Image segmentation in particular turned out to be a major disappointment in this regard – none of the many segmentation algorithms were able to partition a image into anything corresponding to objects. In fact, it was argued (rightly) that low/mid-level segmentation can't possibly be expected to know where one object ends and another one begins since, without object recognition, it doesn't know what objects are!

Meanwhile, the last decade saw an explosion of work in object recognition, most of it without any use of segmentation. Breakthrough results have been achieved in face detection [12, 18] using an exhaustive sliding window approach to predict the presence of a face at every location and scale in the image. A number of researchers have looked at using various visual features directly for recognition, either with a texture-like "bag-of-words" model, or using some spatial relationships (see [9] for an overview). These approaches have shown surprisingly good performance on a number of tasks, including over 65% recognition on the Caltech dataset of 101 object classes. Does this mean that, segmentation, even if it was more accurate, has nothing to contribute to object recognition?

Figure 1: Methods representing objects by bounding boxes often find that up to half of the pixels don't belong to the object of interest (examples from PASCAL Challenge dataset [8]).

1.1 A case for better spatial support

When one considers carefully the recent successes in object recognition, a more nuanced picture emerges. It appears that the current methods are extremely good at a few selected recognition tasks, but quite miserable at most others. For instance, classic rectangular sliding window approaches are known for outstanding results on faces [12, 18], pedestrians [3], and front/side views of cars – all rectangular-shaped objects, but not much else. It seems likely that in cases when the bounding box doesn't cover an object well (e.g. Figure 1), window-based approaches have trouble distinguishing foreground from background.

On the other hand, feature-based methods demonstrate remarkable performance recognizing up to a hundred complicated object categories (grand pianos, anchors, umbrellas, etc) in the Caltech-101 dataset. However, with a single object per image (large and nicely centered), and relatively correlated backgrounds (airplanes in the sky, people in offices, etc) the problem is really more of image classification than object detection. Therefore, it has so far been a challenge to extend detectors trained on Caltech dataset to perform well on novel, cluttered data.

The 2006 PASCAL Visual Object Classes Challenge [8] offers a more realistic dataset with a few large-scale objects per image, although with only 10 object classes. The challenge consists of two tasks: image classification (does this image contain one or more objects of a given class?) and object detection (find and localize all instances of an object class). Tellingly, the contest results on the classification task were overwhelmingly good, especially using the bag-of-words techniques, whereas the performance on the detection task was quite poor. This disparity suggests that the current methods are having a hard time grouping image evidence into coherent objects. Therefore, it is unlikely they will be successful in a setting where a large number of object classes (like in Caltech-101) is presented in realistic images with multiple objects per image (like in PASCAL).

1.2 The Return of Segmentation?

It seems apparent that what is missing from the above recognition efforts is a way to define better spatial support for objects in the image – exactly what the mythical "segmentation stage" is supposed to supply. If we did have perfect spatial support for each object, this would presumably make the recognition task easier in a number of ways. First, by including only the features *on the object* into whatever statistical learning method is being used should greatly reduce the amount of noise in the data, since the relevant features are

not being contaminated by the irrelevant and conflicting ones. In fact, it is possible that the features themselves could be made much simpler, since less burden is being placed on them. Additionally, the shape of the object boundary can also be utilized for recognition. Finally, for general image parsing, the information about object-to-object boundaries is extremely useful not just for resolving object ambiguity via context, but for better modeling of the structure of the entire scene.

But how does one get good spatial support? Researchers have long realized that the original segment-then-recognize paradigm is flawed and that one must consider the two in tandem. As early as 1970, several papers developed image interpretation frameworks that combined bottom-up segmentation with top-down semantic knowledge [15, 7]. Recent efforts in that direction include work by Tu et al. [16], Yu and Shi [20], Bornstein et al. [1], and Shotton et al. [14, 19] among others.

Another direction, taken by Hoiem et al. [6] and Russell et al. [11], is to sample *multiple* segmentations from an image, treating them as hypotheses for object support rather than a single partitioning of the image. The motivation is that, while none of the segmentations are likely to partition the image correctly, *some* segments in *some* of the segmentations appear to provide good spatial support for objects. In this setting, the goal of recognition is no longer finding and localizing objects, but merely scoring the various segments by how likely they are to contain a particular object. The multiple segmentation approach is appealing because of its simplicity and modularity – segmentation and recognition are working in tandem, and yet don't need to know anything about each other, so almost any algorithm could be used for each. However, the approach has yet to be thoroughly evaluated, so it is not clear how much benefit it would provide in the context of finding objects. One of the contributions of this paper is providing just such an evaluation.

1.3 Our Aims

In this paper we will address some of the issues raised above by investigating the potential benefits of using segmentation to find better spatial support for objects. The aim of the paper is to answer the following two questions concerning the role of segmentation in recognition: 1) *Does spatial support matter?* That is, do we even need to segment out objects or are bounding boxes good enough for recognition? 2) *Can segmentation provide better spatial support for objects?* That is, even if spatial support is important, can segmentation deliver it?

1.4 Our Dataset

In this paper, we utilize the Microsoft Research Cambridge (MSRC) dataset [14] of 23[1] object classes. To our knowledge, this is the only object recognition dataset with dense labeling (almost every pixel in each image is labeled) and a large number of object categories. Therefore, this is a good dataset for evaluating segmentation algorithms as well as studying multi-class recognition techniques. To make it more suitable for evaluating segmentation performance, we also manually refined many object boundaries and eliminated "void" regions between objects. On the down side, the small size of the dataset (only 591 images) means that some categories contain only a handful of examples making training object detectors extremely difficult.

[1] Like in previous studies, the horse and mountain categories were removed for recognition purposes due to the small number of instances in each category

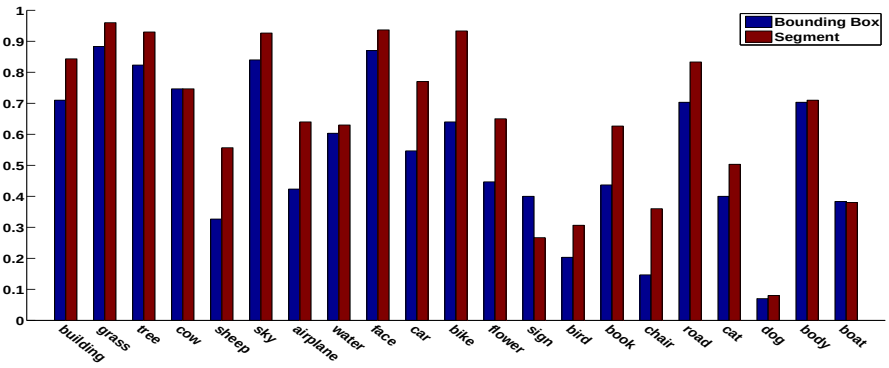


Figure 2: Is good spatial support really worth it? Here, we compare the recall for 21 object classes in the MSRC dataset. For each class, the left bar represents the performance of using ground-truth segments and the right bar is for bounding boxes. In most cases, using pixel-wise segmentations masks gives us a reasonable improvement in classification accuracy.

2 Does spatial support matter?

The first question that must be addressed is whether finding a good spatial support for objects is even a worthwhile goal. After all, there have been suggestions that a bounding box may be able to provide some degree of context and may actually be beneficial [21]. To evaluate this question, we designed a simple experiment on ground-truth segmented objects using a standard object recognition technique. For each object in the dataset, we estimate its class label given a) only the pixels inside the object's ground-truth support region, and b) all pixels in the object's tight bounding box. Because the MSRC dataset is so small, we have chosen the Boosted Decision Tree classifier approach of Hoiem et al. [6] as it appears to perform well under limited amounts of training data. We use exactly the same features as [6] (which measure texture, shape, location, and geometry) extracted from either the ground-truth segment or the tight bounding box of each object.

The results of the experiment, on a per-class basis, can be seen in Figure 2. The first thing to notice is that for 18 out of 21 object categories, the use of correct spatial support improves recognition results. Objects that are poorly approximated by rectangles, such as *sheep*, *bike*, and *airplane*, see the largest improvement (over 50%). Interestingly, recognition on *cars*, which have traditionally been modelled as rectangular, also improves substantially, probably due to the 45 degree views. Categories that don't show improvement with better spatial support are usually doing very well already, e.g. *cows, body*. One exception is *dogs* which is an extremely difficult object class, for which the current methods don't seem to be powerful enough, regardless of the spatial support. Overall, the recognition performance using ground-truth segments is 15% better than using the bounding boxes, increasing from .665 for bounding boxes to .765 for segments. Clearly, the issue of correct spatial support is very important for recognition and should not be ignored. The main question now is, can we actually obtain good spatial support in practice?

3 Can segmentation provide better spatial support?

We have demonstrated that using perfect spatial support can improve recognition, but how can we hope to obtain a good level of spatial support in a purely bottom-up fashion? While there exist many general purpose segmentation engines, studies have shown that none are particularly good at segmenting out individual objects [17, 5]. Instead, we will follow [6, 11] in using multiple different segmentations of the same image, in a way sampling the space of segmentations. In order to put various segmentation schemes on an equal footing, we simply treat the output of any region-generating process as returning a "soup of segments" $\Omega(I)$ for a particular image I.

$$\Omega(I) = (N, \{S_i\}_{i=1}^{N}) \tag{1}$$

When evaluating a segment S's degree of spatial support with respect to a ground truth region G, we compute a normalized overlap score $OS(S,G) \in [0,1]$ (we refer to this overlap score as the spatial support score). When evaluating a soup of segments with respect to G, we report the Best Spatial Support (BSS) score. The BSS is the maximum overlap score in the soup of segments and measures how well the *best* segment covers a ground-truth region.

$$OS(S,G) = \frac{|S \cap G|}{|S \cup G|} \tag{2}$$

For any ground truth region G and soup of segments $\Omega(I)$, we characterize the soup of segments by 2 numbers – the BSS score as well as the number of segments in the soup N. In theory, we want to obtain a small soup (low N) with very high spatial support scores. The performance of a segmentation algorithm across an entire dataset is obtained by averaging the BSS score across each ground truth region in the dataset. Interestingly, under this formulation any sliding window approach can also be seen as generating a soup of segments – namely overlapping rectangular regions at various scales (and possibly orientations).

3.1 The Segmentation Algorithms

In order to have a broad evaluation of bottom-up segmentation, we choose three of the most popular segmentation algorithms, Normalized Cuts (NCuts) [13], the Felzenszwalb and Huttenlocher (FH) algorithm [4] and Mean-Shift [2], to generate our "soup of segments". Following [11], we generate multiple segmentations by varying the parameters of each algorithm. For Normalized Cuts[2], we generate a total of 33 different segmentations per image by varying the number of segments $k = 3,6,9,12,15,18,21,24,27,30,50$ and the image scale to be 100%, 50%, 37% of the original image size. For the FH algorithm, we get 24 segmentations by letting $\sigma = .5,1,1.5,2$, $k = 200,500,1000$, and $min_range = 50,100$. For the Mean-Shift segmentation, we get 33 segmentations, by fixing $min_area = 500$ and varying $spatial_band = 5,7,9$ and $range_band = 1,3,5,7,9,11,13,15,17,19,21$. The number of segmentations for each algorithm was chosen such that size of the multiple segmentation soup was roughly the same.

However, even with multiple segmentations, it's not always possible to get an object as a single segment. Using the intuition that complicated objects often get over-segmented, we will also consider a larger collection of segments by merging up to 3 segments. To get contiguous segments, we only allow segments that are adjacent and come from the same

[2]The version of NCuts we use utilizes the intervening contour cue on top of the Berkeley Pb operator.

segmentation to be merged. Of course, the improvement in the spatial support is done at the cost of considerably increasing the size of the segment soup.

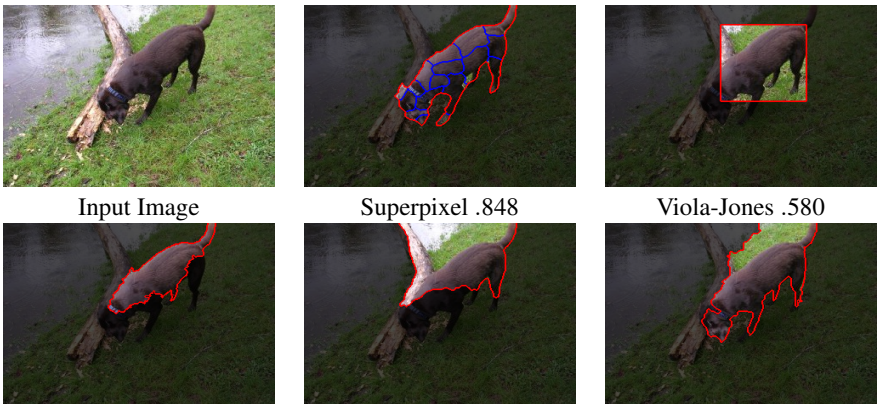
As a baseline, we also consider the soup of segments generated by Viola-Jones [18] style sliding window approach, measuring the best spatial support provided by a square window. Following [18], we choose the smallest window to be 24×24 pixels, and scan each image at 10 scales at $1.25\times$ magnification. The best segments from each algorithm for one example object are shown in Figure 3.

3.2 Experiments

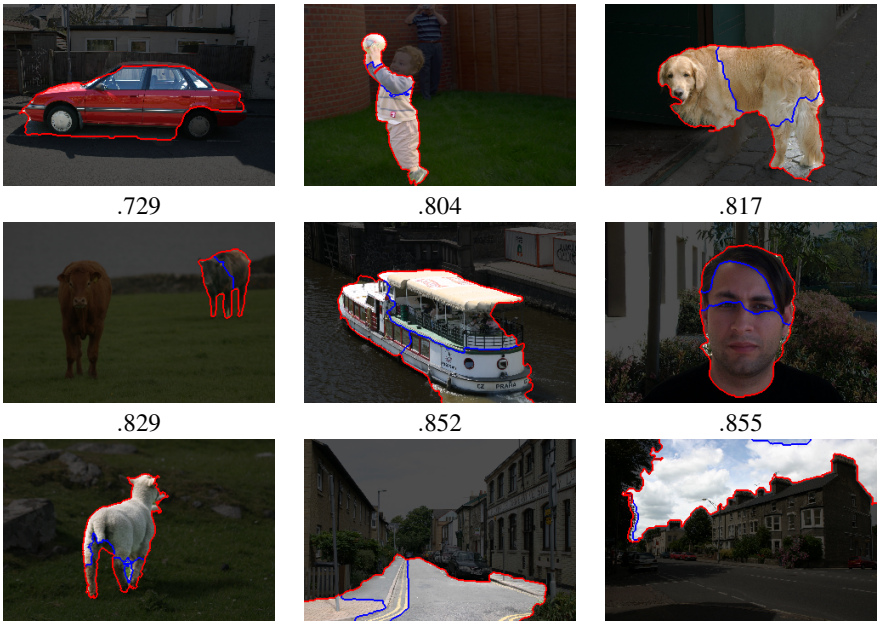
First, we measure the benefits of using multiple segmentations over a single segmentation for the 23 object classes in the MSRC dataset. For each segmentation algorithm, we rank each of the multiple segmentations by their mean BSS score and find the best single segmentation. We then compare the segment soups generated by the best single segmentation, multiple segmentations, multiple segmentations with 1 merge, and multiple segmentations with 2 merges. Each algorithm is evaluated on a per-category basis and compared to Viola-Jones sliding windows (see Figure 4 for results using Mean-Shift). We find that for all algorithms considered, multiple segmentations drastically outperform the best single segmentation. Adding a single merge to the soup of segments also significantly improves the BSS scores, and we notice diminishing returns when adding more merges. We also create a superpixel-based upper bound (the SP Limit) on how well we expect a bottom-up segmentation algorithm to cover objects of a particular class. For the superpixel limit, we create superpixels [10] by over-segmenting each image into approximately 200 regions using Normalized Cuts and finding the optimal spatial support for each ground-truth object instance with respect to the superpixel map (example shown in Figure 3). The SP Limit is just the average performance of superpixels. To no surprise, the multiple segmentation approach is able to provide very good spatial support for object categories whose instances have a relatively homogeneous appearance such as grass, cow, and road while there is still plenty of room left for improving spatial support for complicated objects such as airplanes and boats.

This type of analysis suggests that merging adjacent segments always helps since we only consider the *best* spatial support score. However, in order give a fair evaluation we must also characterize each segment soup by its size. Thus, we also compare the mean BSS of each algorithm versus segment soup size, as seen in Figure 5 (here we use log segments, since the size of the soup quickly gets out of hand). For each of the three algorithms, we compute the mean BSS and the mean soup size for the single best segmentation, multiple segmentations, multiple segmentations with 1 merge, and multiple segmentations with 2 merges. While both FH and Mean-Shift have similar average performance, they both significantly outperform NCuts. To complement the superpixel upper bound, we also determine an upper-bound for the overlap score if using tight bounding boxes (the BB Limit). This is computed by finding the rectangular window with the best overlap score for each object instance in the MSRC dataset and averaging those overlap scores. (This is merely a limit, since in practice it is intractable to slide rectangular regions of *all* aspect ratios and *all* scales across an image.)

We also consider what happens when we concatenate the best single segmentation from each algorithm, the multiple segmentations from each algorithm, and so on. Both FH and Mean-Shift were only able to approach the BB Limit by considering the largest soup of segments (created from multiple segmentations with 2 merges); however, by concatenating the three different algorithms we were able to surpass the BB Limit with a much smaller number of segments. This suggests that the different segmentation algorithms are



(a)



(b)

Figure 3: (a) Best overlap using a single segmentation. Under each image is the segmentation algorithm from which the segment came from as well as the corresponding overlap score. Segment boundaries are represented in red and boundaries of merged segments are represented in blue. The first row of (a) shows the original input image, a superpixel approximation to the ground truth region, as well as the square with best spatial support (Viola-Jones). (b) Nine example results when using a soup of approximately 30k segments. For each object we display the segment with the best spatial support. Here we consider the segment soup created by concatenating all three segmentation algorithms and merging up to 2 segments.

complementary, with each algorithm providing better spatial support for different object categories.

We also note that when using Viola-Jones style window sliding, we generate on the average an order of magnitude more regions than the largest segment soup (created from the combination of the 3 algorithms and up to 2 merges). In addition, the performance of such a square-window based approach is not only far from the BB Limit, but significantly inferior to the multiple segmentation approach. Quantitatively, using Viola-Jones approach we obtain a mean BSS score of .426 while combining the output of the three segmentation algorithms with 2 merges gives a mean BSS of .855 – which is a 100% improvement in spatial support over Viola-Jones window sliding while considering an order of magnitude less segments.

4 Discussion

In this paper, our central goal was to carefully examine the issues involved in obtaining good spatial support for objects. With segmentation (and multiple segmentation approaches in particular) becoming popular in object recognition, we felt it was high time to do a quantitative evaluation of the benefits and the trade-offs compared to traditional sliding window methods. The results of this evaluation can be summarized in terms of the following "take-home" lessons:

Correct spatial support is important for recognition: We confirm that knowing the right spatial support leads to substantially better recognition performance for a large number of object categories, especially those that are not well approximated by a rectangle. This should give pause to researchers who feel that recognition can be solved by training Viola-Jones detectors for all the world's objects.

Multiple segmentations are better than one: We empirically confirm the intuition of [6, 11] that multiple segmentations (even naively produced) substantially improve spatial support estimation compared to a single segmentation.

Mean-Shift is better than FH or NCuts, but together they do best: On average, Mean-Shift segmentation appeared to outperform FH and NCuts in finding good spatial support for objects. However, for some object categories, the other algorithms did a better job, suggesting that different segmentation strategies are beneficial for different object types. As a result, combining the "segment soups" from all three methods together produced by far the best performance.

Segment merging can benefit any segmentation: Our results show that increasing the segmentation soup by merging 2 or 3 adjacent segments together improves the spatial support, regardless of the segmentation algorithm. This is because objects may contain parts that are very different photometrically (skin and hair on a face) and would never make a coherent segment using bottom-up strategies. The merging appears to be an effective way to address this issue without doing a full exhaustive search.

"Segment soup" is large, but not catastrophically large: The size of the segment soup that is required to obtain extremely good spatial support can be quite large (around 10,000 segments). However, this is still an order of magnitude less than the number of sliding windows that a Viola-Jones-style approach must examine. Moreover, it appears that using a number of different segmentation strategies together, we can get reasonable performance with as little as 100 segments per image!

In conclusion, this work takes the first steps towards understanding the importance of providing good spatial support for recognition algorithms, as well as offering the practitioner a set of concrete strategies for using existing segmentation algorithms to get the best object support they can.

References

[1] Eran Borenstein, Eitan Sharon, and Shimon Ullman. Combining top-down and bottom-up segmentation. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, page 46, Washington, DC, USA, 2004.

[2] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Patt. Anal. Mach. Intell.*, 24(5):603–619, 2002.

[3] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proc. CVPR*, 2005.

[4] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *Int. Journal of Computer Vision*, 59(2):167–181, 2004.

[5] Feng Ge, Song Wang, and Tiecheng Liu. Image-segmentation evaluation from the perspective of salient object extraction. In *CVPR*, 2006.

[6] D. Hoiem, A. Efros, and M. Hebert. Geometric context from a single image. In *Proc. Int. Conf. Comp. Vision*, 2005.

[7] Y. Ohta, T. Kanade, and T. Sakai. An analysis system for scenes containing objects with substructures. In *IJCPR*, pages 752–754, 1978.

[8] The pascal object recognition database collection. Website, 2006. http://www.pascal-network.org/challenges/VOC/.

[9] J. Ponce, M. Hebert, C. Schmid, and A. Zisserman. *Toward Category-Level Object Recognition*. Springer-Verlag Lecture Notes in Computer Science, 2006. In press.

[10] Xiaofeng Ren and Jitendra Malik. Learning a classification model for segmentation. In *International Conference on Computer Vision*, pages 10–17, 2003.

[11] Bryan C. Russell, Alexei A. Efros, Josef Sivic, William T. Freeman, and Andrew Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *Proc. CVPR*, 2006.

[12] H. Schneiderman and T. Kanade. A statistical method for 3D object detection applied to faces and cars. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, Hilton Head, SC, 2000.

[13] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. PAMI*, 22(8):888–905, August 2000.

[14] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, 2006.

[15] J.M. Tenenbaum and H.G. Barrow. Experiments in interpretation guided segmentation. *Artificial Intelligence*, 8(3):241–274, June 1977.

[16] Z. Tu, X. Chen, A. Yuille, and S.-C. Zhu. Image parsing: unifying segmentation, detection, and recognition. In *Proc. Int. Conf. Comp. Vision*, volume 1, pages 18–25, 2003.

[17] Ranjith Unnikrishnan, Caroline Pantofaru, and Martial Hebert. A measure for objective evaluation of image segmentation algorithms. In *Proceedings of the 2005 CVPR Workshop on Empirical Evaluation Methods in Computer Vision*, June 2005.

[18] P. Viola and M.J. Jones. Robust real-time face detection. *Int. J. of Comp. Vision*, 57(2):137–154, 2004.

[19] John Winn and Jamie Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *CVPR*, pages 37–44, 2006.

[20] Stella X. Yu and Jianbo Shi. Object-specific figure-ground segregation. In *CVPR*, 2003.

[21] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classifcation of texture and object categories: A comprehensive study. *Int. J. of Comp. Vision*, 2007. in press.

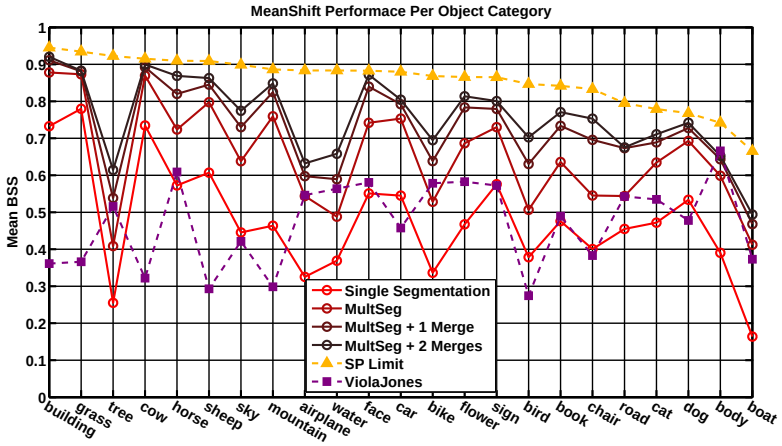


Figure 4: Performance across object category for the Mean-Shift algorithm. The performance of the segment soup created by the single best segmentation, multiple segmentations, multiple segmentations with 1 merge, and multiple segmentations with 2 merges is compared to a Viola-Jones window sliding approach as well as a superpixel-based uppper-bound on segmentation performance.

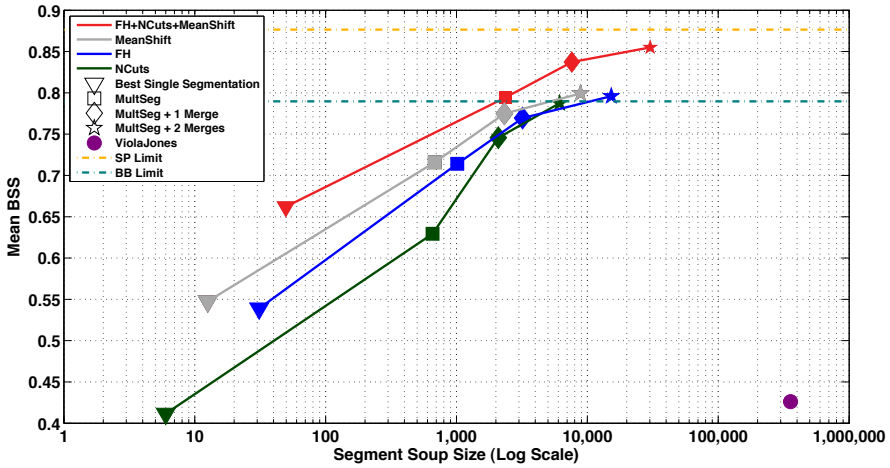


Figure 5: Mean BSS versus Segment Soup Size for different segment soups generated from each of the 3 segmentation algorithms compared to Viola-Jones window sliding. A log scale is used along the x-axis since segment soups get large very fast. A combination of the 3 algorithms is also evaluated as well as a superpixel-based upper bound and a bounding box upper bound on performance.

Delving into the whorl of flower segmentation

Maria-Elena Nilsback and Andrew Zisserman
Visual Geometry Group
Deparment of Engineering Sciene, University of Oxford

Abstract

We describe an algorithm for automatically segmenting flowers in colour photographs. This is a challenging problem because of the sheer variety of flower classes, the intra-class variability, the variation within a particular flower, and the variability of imaging conditions – lighting, pose, foreshortening etc.

The method couples two models – a colour model for foreground and background, and a generic shape model for the petal structure. This shape model is tolerant to viewpoint changes and petal deformations, and applicable across many different flower classes. The segmentations are produced using a MRF cost function optimized using graph cuts.

The algorithm is tested on 13 flower classes and more than 750 examples. Performance is assessed against ground truth segmentations.

1 Introduction

There is an interesting research theme in computer vision of using a general class model to initialize a segmentation, and then improving the segmentation using image specific features. The LOCUS approach of Winn and Jojic [15] is a good example, where a shape matte (e.g. for a car or horse) is used to propose a foreground/background segmentation, colour distributions are then gathered in each region, and the final segmentation achieved using a colour based binary MRF optimized with graph cuts. The `ObjCut` algorithm of Kumar *et al.* [5] similarly proposes a position and configuration for a class instances (e.g. a cow or horse) using a pictorial structure with boundary shapes and texture features, and then, again, incorporates colour measured from the proposed regions to carry out a MRF segmentation with graph cuts. Similar ideas are present in several other recent class based segmentation methods [2, 7, 8, 13]. Even though these methods are applied to a variety of classes, e.g. cars, cows, faces, horses, for the most part a different model is used for each class, and a different model is used for each view of the class (e.g. cars rear, cars side).

In this paper we introduce two variations on this theme: first, we reverse the order in which the features are used – we start with colour to propose a foreground/background segmentation and use this to initialize image specific shape measurements; second, we use a generic shape model which is applicable across a number of classes and viewpoints.

Our target application is segmenting flowers from photographs as part of the process of automated flower classification [11]. In the manner of `ObjCut` [5] our goal is to automatically segment out the flower given only that the image is known to contain a flower, but no other information on the class or pose. Figure 1 illustrates the challenge of the segmentation task, and shows the fitted generic flower shape model for several flower classes.

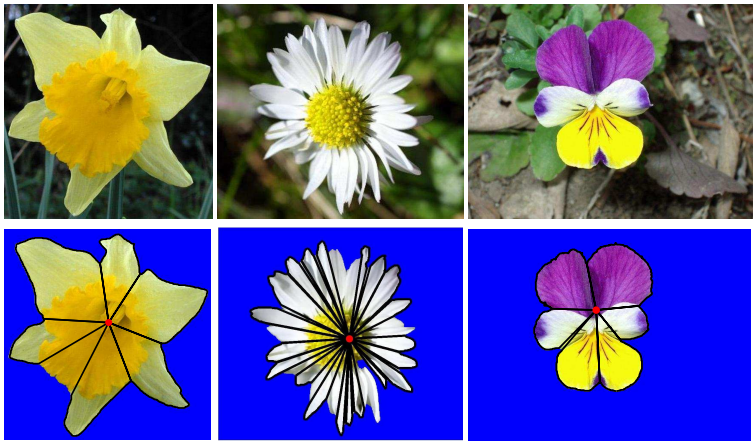


Figure 1: Top row: typical flower photographs, note the variety of imaging conditions. The classes are (left to right) daffodil, daisy and pansy. Bottom row: the flower shape model fitted automatically (note the lines to the flower "centre") and the resulting segmentation. The same shape model is used in all cases despite differing numbers of whorls, the large variation in the number of petals (daffodil & pansy vs daisy), and variation in the shape of the petals.

The method is evaluated on the Oxford Flower Dataset available at [1]. The dataset has 17 flower classes (e.g. buttercup, daffodil, iris, pansy), with photographs exhibiting typical (large) variations in viewpoint, scale, illumination and background. Segmenting such photographs is challenging due to both the variety of colours and the variety of shapes. If we knew that we were looking for a daffodil or a bluebell, we could build one foreground and one background colour model for each of these classes – though this would still give problems with the sky being segmented as foreground in a bluebell photograph for example. But here we do not wish even to specify the flower class in advance. Shape also poses a challenge because of the many different types and whorls of petals. Even on the same flower there are local deformations of the petal shape.

Previous work on segmenting flowers using colour by Das *et al.* [4] has exploited domain knowledge on the precise colour of flowers. For example, a flower is rarely brown or green (this is not valid for the sunflower class in the Oxford Flower Dataset). In addition [4] learn an image specific background colour model from the periphery of the image. Again, this method is not applicable in our case since the flowers often reach the image periphery and so the background model would be corrupted with foreground colours. Saitoh *et al.* [12] have also proposed a method for extracting flowers regions. It is based on "Intelligent Scissors" [10], which find the path between two points that minimizes a cost function dependant on image gradients. The method works under the assumption that the flower is in focus and in the centre of the photograph and that the background is out of focus. Under this assumption the cost between any two points on the flower is smaller than the cost between a point in the background and a point in the foreground. By fixing the midpoint of the image as part of the flower this can be used as a starting point for finding the flower region. This method requires no prior colour information. Unfortunately these assumptions do not apply in our case.

This paper is organized as follows. In Section 2 we describe our segmentation algorithm, and introduce the flower shape model. Section 3 specifies the experimental procedure, and Section 4 assesses the quality of the flower segmentations against ground truth.

2 The Segmentation Algorithm

2.1 Overview

We first obtain an initial flower segmentation using *general* (non-class specific) foreground and background colour distributions. These distributions are learnt by labelling pixels in a few sample images of each class in the dataset as foreground (i.e. part of the flower), or background (i.e. part of the greenery), and then averaging the distributions across all classes. Given these general foreground and background distributions, a binary segmentation is obtained using the contrast dependent prior MRF cost function of [3], optimized with graph cuts. This is the method used in [11]. This segmentation may not be perfect, but is often sufficient to extract at least part of the external boundary of the flower.

The generic flower shape model is then fitted to this initial segmentation in order to detect petals. The model selects petals which have a loose geometric consistency using an affine invariant Hough like procedure. The image regions for the petals deemed to be consistent are used to obtain a new *image specific* colour foreground model. This image specific foreground model replaces the general foreground model (the background model is unchanged), and the MRF segmentation is repeated. In cases where the initial segmentation was not perfect, the use of the image specific foreground often harvests more of the flower. The steps of shape model fitting and image specific foreground learning can then be iterated until convergence.

The algorithm is illustrated in figure 2. We first describe these stages in more detail and then give implementation details.

2.2 Segmentation

The segmentation method is an implementation of the binary MRF with a contrast dependent prior proposed by [3]. Each pixel has two possible states corresponding to foreground or background. The probability of each state depends on the likelihood that the pixel's colour corresponds to the foreground or background colour distribution. The cost function that is minimized has a unary term for each pixel (corresponding to the log likelihood of its state) and a pairwise prior which adds a cost if there is a difference in the state between a pixel and its neighbour, *but* this cost is diminished if there is a strong gradient between the pixels (as measured in the images). The inclusion of data-dependent pairwise terms for pixels in a clique gives a substantial improvement in segmentation quality, and the resulting MRF can still be minimized using graph cuts as described in [3].

2.3 Generic flower shape model

The flowering parts of a flower can be either petals, tepals or sepals. For simplicity we will refer to these as petals.

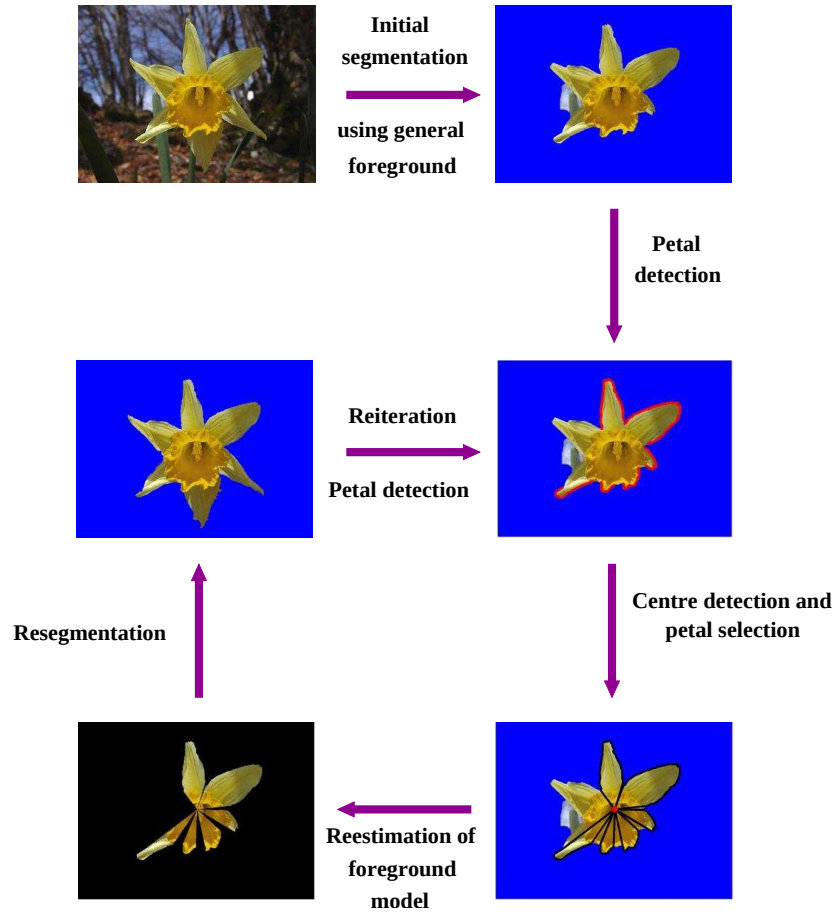


Figure 2: An overview of the segmentation algorithm.

The geometric representation of a petal and flower is illustrated in figure 3. In a frontal view of a "perfect" flower (think of a daffodil) the lines joining the extreme point to the mid-point of each petal coincide at the flower centre. However, under reasonable viewpoints, and provided the deformation of the flower is not excessive, this construction is still a good approximation, and the agreement of each petal with a common flower centre can be used to verify or remove putative petals. The model represents a loose geometric configuration of petals. The particular shape of the petals and their number is not specified, nor is the symmetry of their arrangement. The only requirement is that the image regions deemed to be petals should agree on the flower "centre".

The construction of the extreme point is based on the canonical frame representation of Lamdan *et al.* [6]. Note, this construction (and indeed the entire flower shape model) is affine invariant. So, for example, it is unaffected if the image is rotated, scaled, or its aspect ratio changed. In particular this degree of invariance makes the construction

tolerant to out of plane rotations – as the flower head turns from the camera.

Since the model does not make strong requirements on the number or arrangement of petals, it is applicable to flowers with a small number or petals with rotational symmetry (e.g. daffodils, windflowers), to flowers with very many petals with rotational symmetry (e.g. sunflowers, dandelions), and to flowers with a small number of petals without rotational symmetry (e.g. iris, pansies). Somewhat surprizingly, the same construction may be used for side views of flowers as illustrated by the fritillaries and tiger lilies in figure 5.

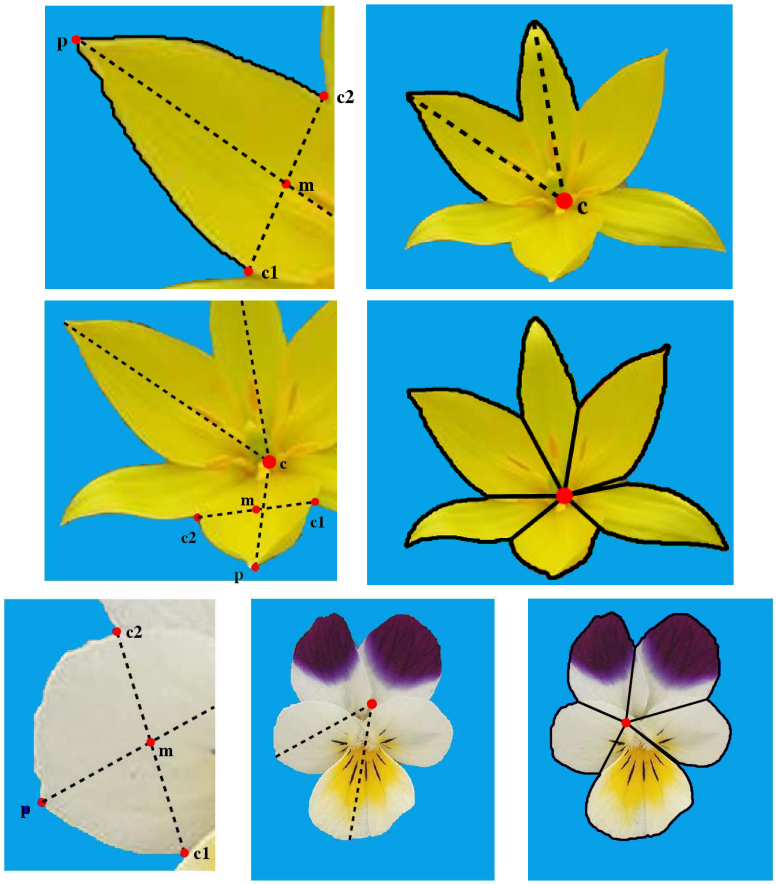


Figure 3: **The geometric flower model.** Top row: (left) A petal is defined by the boundary curve between the corners $\mathbf{c}_1$ and $\mathbf{c}_2$. The extreme point $\mathbf{p}$ is the point on the curve furthest from the line $\langle \mathbf{c}_1, \mathbf{c}_2 \rangle$, and with tangent parallel to the line. (right) a hypothesis for a centre obtained by intersecting two petal mid lines $\langle \mathbf{p}, \mathbf{m} \rangle$. Middle row: (left) The consistency of a putative centre $\mathbf{c}$ is measured by the deviation of the line $\langle \mathbf{c}, \mathbf{p} \rangle$ from the mid-point $\mathbf{m}$, and (right) the petal regions consistent with the centre with greatest consistency. Bottom row: A second example of fitting the model, here for a pansy, left to right: single petal model, putative centre, final centre and verified petals consistent with the centre.

2.4 Implementation details

Colour distribution representation and likelihood. The colour distribution is represented in RGB-space using a histogram with $N = 14^3$ bins, i.e. 14 bins per dimension. The log likelihood of a pixel under the normalized histogram $hist(r,g,b)$ is then proportional to $-\log(hist(r,g,b))$. Empty bins are dealt with by adding $0.5/N$ to all bins (including non-empty ones) and then renormalizing the histogram. This avoids the problem of computing the log of empty bins, and is equivalent to a uniform prior over the colour space [9].

Petal detection. The boundary of the segmented foreground region is traced by examining an 8-neighbourhood around each foreground pixel. Once all the boundary pixels are found, corners are detected using a "worm" that slides around the boundary and declares a corner at a boundary point where the distance to the straight line between its head and tail exceeds a threshold, and is a local maximum. Convex corners (wrt to the background region) are the potential petal corners.

The top row in figure 3 shows how petals are detected. Given two neighbouring corner points, $\mathbf{c}_1$ and $\mathbf{c}_2$, the line $\langle \mathbf{c}_1, \mathbf{c}_2 \rangle$ is determined. Next we search for a point, $\mathbf{p}$, on the boundary between $\mathbf{c}_1$ and $\mathbf{c}_2$, which has the same orientation as the line $\langle \mathbf{c}_1, \mathbf{c}_2 \rangle$. If no points are found or if the ratio between the distance along the boundary and the length of $\langle \mathbf{c}_1, \mathbf{c}_2 \rangle$ is smaller than a threshold, t, the petal between the two corners is removed. The line $\langle \mathbf{p}, \mathbf{m} \rangle$ joining $\mathbf{p}$, and the middle point $\mathbf{m}$ of the segment $\langle \mathbf{c}_1, \mathbf{c}_2 \rangle$ indicates where the centre of the flower might lie.

Centre voting. For each pair of petals we find the intersection between the corresponding lines $\langle \mathbf{p}, \mathbf{m} \rangle$. This point of intersection, $\mathbf{c}$, is a putative centre. Additional petals are added with a cost corresponding to deviation from $\mathbf{m}$ of the intersection, $\mathbf{i}$, of the lines $\langle \mathbf{c}_1, \mathbf{c}_2 \rangle$ and $\langle \mathbf{p}, \mathbf{c} \rangle$ (the cost is measured on each petal). The deviation is normalized by the half-length of the segment $\langle \mathbf{c}_1, \mathbf{c}_2 \rangle$. If the deviation is greater than unity, i.e. the intersection $\mathbf{i}$ lies outside the segment, then the petal is not added to the putative centre. The centre with highest consistency, in terms of number of petals and lowest cost, is chosen to be the centre of the flower. The petals consistent with this centre, together with the triangular wedge of the centre region, are used to refine the foreground model. If no centre is found we lower the threshold, t, in order to detect more petals.

Segmentation convergence. The detection of petals and refinement of the foreground model is repeated iteratively. Once there is no change in the segmented region between two consecutive iterations the algorithm has converged. If the algorithm does not converge, we choose the initial segmentation.

Parameter selection The size of the worm and the threshold, t, are learnt from a subset of 10 images per category by optimizing the overlap score.

3 Experimental procedure

For the experiments we exclude images in the Oxford Flower Dataset that contain fields of flowers or very small flowers, that are too subsampled, or where there is no clear foreground region. Following these criteria, four of the classes (snowdrops, lily of the valley's, cowslips and bluebells) have insufficient images and are removed. This leaves 13 flower classes with a total of 753 images.

All of the remaining images were ground truth labelled into foreground and background regions using a trimap. In the trimap a small proportion of pixels and certain regions are left unlabelled – principally the pixels very close to the flower boundary, and regions that are difficult to classify, e.g. a small out of focus flower in the background. Not labelling pixels close to the boundary greatly simplifies the annotation procedure as precisely selecting pixels near a boundary is painfully slow. Due to the contrast dependent prior the graph cuts algorithm generally aligns its boundaries with image edges, so we are more interested in assessing mistakes elsewhere. Figure 4 shows an example trimap labelling for an image. The red area is labelled foreground, the green area background, and the black area is unlabelled. The trimaps are available at [1].

The performance is measured by computing an overlap score, P, between the ground truth segmentation and the segmentation obtained by the algorithm:

$$P = \frac{true\,foreground \cap segmented\,foreground}{true\,foreground \cup segmented\,foreground} \tag{1}$$

In evaluating this score only the labelled pixels are considered (i.e. the unlabelled pixels have no effect). The score is 1 for a perfect segmentation, and less than 1 otherwise.

Figure 4: An image and its corresponding trimap ground truth labelling. The black region is unlabelled, and is not used in assessing performance. The red region is foreground, and the green region background.

4 Results

Figure 5 shows examples of the coupled algorithm improving the segmentation for different classes. It can be seen that it both manages to retrieve missing flower regions, and remove background regions that initially were misclassified. Typically only a few iterations are required – in 68% of the cases there is very little change after the first iteration. After five iterations 97% of the segmentations have converged.

For a quantitative comparison we plot the proportion of images that have an overlap score greater than P, against P. This graph is chosen to resemble an ROC curve, in that perfect performance corresponds to a curve approaching the top left corner. In figure 6 we compare performance using only the general colour distributions without using the

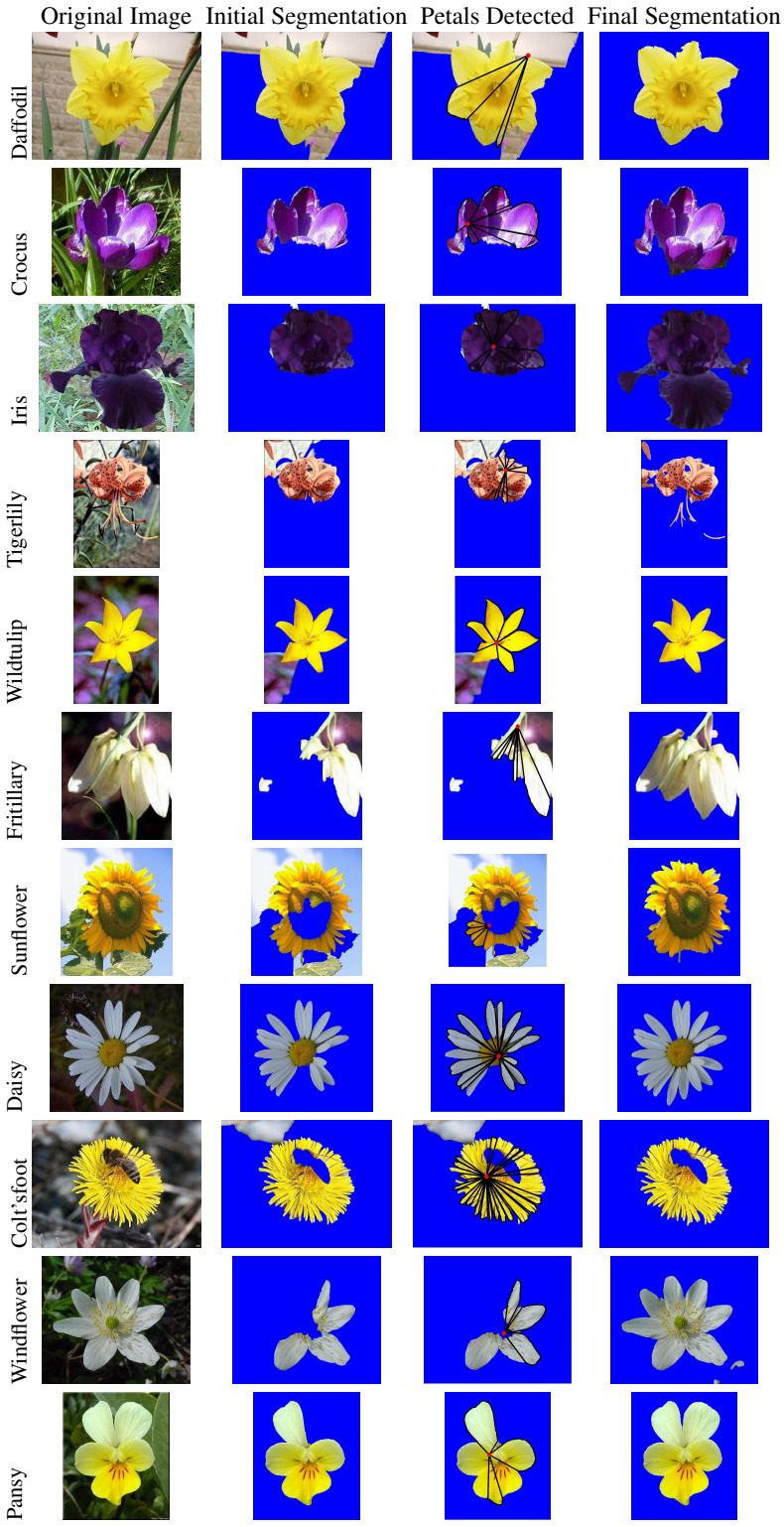


Figure 5: Example results for a number of classes. In each case iteratively updating the foreground colour model leads to a full segmentation, despite only a partial segmentation at the start. There are examples where missed foreground is retrieved (crocus, iris) and where erroneous background is removed (tiger lily, wild tulip).

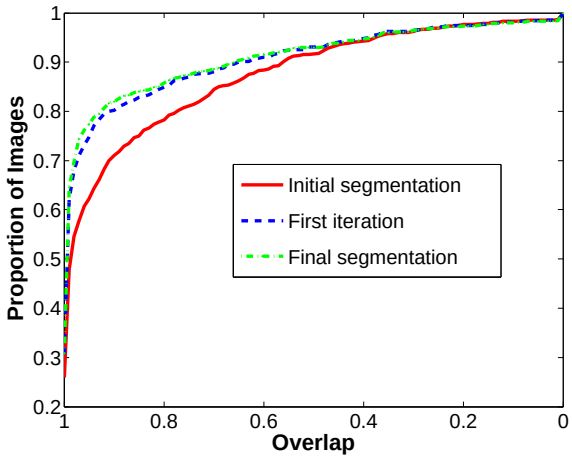


Figure 6: Comparison of segmentation performance with and without using the iterative algorithm. The x axis shows the overlap score P, and the y axis the proportion of images that have an overlap score greater than or equal to P. The solid red line shows the results of the initial segmentation, the dashed blue line the results of the first iteration, and the green line is the result after convergence. Note the clear improvement brought by the iterative algorithm over the initial segmentation.

shape model (essentially the method of [11]), to the performance using the coupled segmentation algorithm (iterating colour and shape fitting). It can be seen that learning an image-specific foreground distribution significantly improves the segmentation: for example the initial segmentation has an overlap of 95% for 62% of the images and this is increased to 77% (with 75% of the images reaching this point after a single iteration).

To give an idea of the type of error that occurs for various overlap errors, figure 7 shows typical segmentation deficiencies at levels of P in the range 0.92 – 1.0 for two different images.

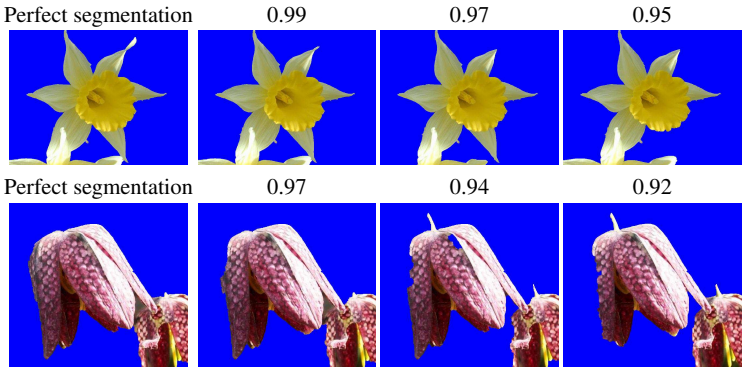


Figure 7: Segmentation deficiencies for different levels of overlap. The overlap score is given above the images. Top row: A bit of the top right petal disappears at 0.99, and at 0.95 an entire petal is lost. Bottom row: At 0.97 a bit of the left edge is lost, at 0.94 a bit of the stalk is added and a hole appears, and at 0.92 part of the left petal is missing.

5 Conclusion

We have demonstrated that flower segmentations can be significantly boosted by using an image specific colour distribution, and that this distribution can be learnt automatically by fitting a general flower shape model.

We have concentrated here on segmenting out the flower. A similar method (general colour model, generic shape model) could now be used for segmenting out the leaves and stalk – making strong use of green in the colour model. The current model is not suitable for fields of flowers. For such cases a model that groups repeated shapes would be more appropriate, such as [14].

The geometric model we have introduced here is a step towards extracting petal structure. The next step is flower classification based on petal shape and configuration.

Acknowledgements

This work was funded by EC Marie-Curie Training Network VISIONTRAIN.

References

[1] http://www.robots.ox.ac.uk/∼vgg/data/flowers/.

[2] E. Borenstein, E. Sharon, and S. Ullman. Combining top-down and bottom-up segmentation. In *CVPRW*, volume 4, page 46, 2004.

[3] Y. Y. Boykov and M. P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *Proc. ICCV*, volume 2, pages 105–112, 2001.

[4] M. Das, R. Manmatha, and E. M. Riseman. Indexing flower patent images using domain knowledge. *IEEE Intelligent Systems*, 14(5):24–33, 1999.

[5] M. P. Kumar, P. H. S. Torr, and A. Zisserman. OBJ CUT. In *Proc. CVPR*, 2005.

[6] Y. Lamdan, J. T. Schwartz, and H. J. Wolfson. Object recognition by affine invariant matching. In *Proc. CVPR*, pages 335–344, 1988.

[7] B. Leibe and B. Schiele. Interleaved object categorization and segmentation. In *Proc. BMVC.*, volume 2, pages 264–271, 2003.

[8] A. Levin and Y. Weiss. Learning to combine bottom-up and top-down segmentation. In *Proc. ECCV*, volume 4, pages 581–594, 2006.

[9] C. Manning and H. Schutze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[10] E. Mortensen and W. A. Barrett. Intelligent scissors for image composition. In *Proc. ACM SIGGRAPH*, pages 191–198, 1995.

[11] M-E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In *Proc. CVPR*, volume 2, pages 1447–1454, 2006.

[12] T. Saitoh, K. Aoki, and T. Kaneko. Automatic recognition of blooming flowers. In *Proc. ICPR*, volume 1, pages 27–30, 2004.

[13] J. Shotton, J. Winn, C Rother, and A. Criminisi. *TextonBoost*: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *Proc. ECCV*, pages 1–15, 2006.

[14] T. Tuytelaars, A. Turina, and L. Van Gool. Noncombinatorial detection of regular repetitions under perspective skew. *IEEE PAMI*, 25(4):418–432, 2003.

[15] J. Winn and N. Jojic. Locus: Learning object classes with unsupervised segmentation. In *Proc. ICCV*, pages 756–763, 2005.

Author Index

Adams, J. E. , 302
Al-Osaimi, F. R. , 132
Alexander, D. C. , 429
ALLEZARD, N. , 341
Amores, J. , 351
Antone, M. , 850
Apostoloff, N. , 509
Arif, M. , 312

Bai, X. , 172
Balasuriya, S. L. , 222
Bartoli, A. E. , 42, 1050
Bauckhage, C. , 590
Bebis, G. , 202
BEGARD, J. , 341
Bennamoun, M. , 132
Berclaz, J. , 690
Berent, J. , 409
Bernier, O. , 680
Beszéde, M. , 479
Bhalerao, A. H. , 312
Bischof, H. , 272, 459, 1080
Bors, A. G. , 640, 710
Boulanger, P. , 840
Bouridane, A. , 419
Bowen, A. , 381
Branch, J. W. , 840
Browne, M. , 331
Bull, D. , 730

Caglioti, V. , 820
Calway, A. D. , 52
Campbell, N. D. F. , 530
Canagarajah, N. , 730
Caputo, B. , 1090
Castellini, C. , 1090
Castle, R. O. , 1100
Chandran, S. , 740
Chaudhuri, S. , 900
Chebrolu, H. , 1070
Chekhlov, D. , 52
Chelberg, D. , 1070
Chen, Z. , 920
Christmas, B. , 650
Cipolla, R. , 530, 790
Clipp, B. , 720
Cockshott, W. P. , 222
Cohn, A. G. , 610
Cooperstock, J. R. , 371
Cootes, T. F. , 302, 880, 1060
Cristinacce, D. , 880
Crookes, D. , 419
Crosier, M. S. , 540
Culverhouse, P. F. , 479

Dambreville, S. , 520
Damen, D. , 72
Daniilidis, K. , 292
De Lange, D.-J. , 439
Dekeyser, F. , 950
Den Hollander, R. , 439, 860
Dhome, M. , 950
Di Stefano, L. , 690
Ding, Y. , 162
Donner, R. , 1080
Doshi, A. , 640
Dragotti, P. L. , 409
Durrant-Whyte, H. , 232

Efros, A. A. , 560
Eggert, J. , 12
El-Baz, A. S. , 92
Escalante, H. J. , 600
Ess, A. , 361

Fernandez, F. , 630
Fitzgibbon, A. W. , 1120
Fleet, D. J. , 700
Fleuret, F. , 690
Frahm, J.-M. , 720
Frank, O. , 232
Friedman, Y. , 850
Fua, P. , 690
Furukawa, R. , 780

Galasso, F. , 660

Gawley, D. J. , 1100
Gee, A. P. , 52
Gibson, G. J. , 82
Gimel'farb, G. , 92
Giusti, A. , 820
Goncalves, L. , 192
Gong, S. , 242, 469, 550
Gong, s. , 620
Gribben, H. , 331
Griffin, L. D. , 540
Grum, M. , 710
Gueham, M. , 419
Gunawan, I. P. , 222
Guo, W. , 262

Haines, T. S. F. , 910
Hall, P. M. , 122, 172
Han, D. , 262
Hancock, E. R. , 499
Hanjalic, A. , 860
Hebert, M. , 2, 212, 940
Heesch, D. , 930
Hernandez, C. , 530
Hernandez-Marin, S. , 82
Hewitt, R. , 192
Hogg, D. , 72
Hu, B. , 960
Hunter, D. W. , 449
Hyndman, M. , 700

Iwamura, M. , 182, 1010

Jepson, A. , 700
Jermyn, I. H. , 960
Jerome, S. , 770
Jin, Y. , 670
Joshi, M. V. , 760

Kälviäinen, H. , 252
Körner, E. , 12
Kalesnykiene, V. , 252
Kamarainen, J.-K. , 252
Kanatani, K. , 282
Katz, R. , 232
Kauppi, T. , 252
Kawasaki, H. , 780
Kimia, B. B. , 1030
Kise, K. , 182, 1010
Kittler, J. , 650
Klein, G. , 1100
Klingenberg, C. P. , 870
Koch, C. , 1000
Kubicki, M. , 322

Lacey, G. , 389
Lakshmikumar, A. , 1110
Langs, G. , 1080
Lanza, A. , 690
Lasenby, J. , 660
Lecca, M. , 112
Leek, C. , 990
Lensu, L. , 252
Leonardis, A. , 272
Lhuillier, M. , 950
Li, C. , 122
Li, W. , 32, 262
Li, Y. , 399
Li, Z. , 262
Liu, J. , 32, 1070
Liu, Q. , 580
Liu, X. , 399
Liu, x. , 830
Llorca, D. F. , 389
Loke, Y. R. , 980
Loss, L. , 202
Lu, H. , 580
Lukins, T. C. , 142
Luo, J. , 1090
Lutton, E. , 630

Ma, S. , 580
Malcolm, J. , 890
Malisiewicz, T. , 560
Matzka, S. , 750
Mayol, W. W. , 52
McOwan, P. W. , 242, 469
Melonakos, J. , 322
Meng, Y. , 449
Messelodi, S. , 112
Mian, A. S. , 132
Micusik, B. , 1080
Miller, P. , 331

Mills, A. M. , 1060
Mills, S. , 970
Mirmehdi, M. , 1040
Moeller, B. , 152
Mohan, V. , 322
Mokhtarian, F. , 670
Molana, R. , 292
Montes y Gómez, M. , 600
Mouragnon, E. , 950
Mullins, A. , 381
Munich, M. E. , 192
Murray, D. W. , 1100

Naeem, A. , 970
Nakai, T. , 1010
Namboodiri, V. P. , 900
Neubeck, A. , 361
Nicolescu, M. , 202
Niethammer, M. , 322, 520
Nilsback, M.-E. , 570
Noguchi, K. , 182
Noriega, P. , 680

Olague, G. , 630
Olsen, S. I. , 1050
Orabona, F. , 1090
Ozcanli, O. C. , 1030

Palaniswamy, S. , 870
Pantofaru, C. , 212
Patron, A. , 62
Peng, T. , 960
Petillot, Y. R. , 750
Petrou, M. , 930
Petrovic, V. S. , 1060
Peyras, J. , 489
Pietilä, J. , 252
Pollefeys, M. , 720
Posch, S. , 152
Pridmore, T. P. , 970
Prieto, F. , 840
Prinet, V. , 960
Pryor, G. , 102
Pryor, G. D. , 22

Qayyum, Z. U. , 610
QI, Z. , 371

Radeva, P. , 351
Rajpoot, N. , 312, 381
Ranganath, S. , 980
Raninen, A. , 252
Rathi, Y. , 890
Redmill, D. , 730
Rehman, T. u. , 22, 102
Reid, I. , 62, 800, 1120
Roberts, M. G. , 302
Roth, P. M. , 272
Russell, D. M. , 550
Ruta, A. , 399

Sandini, G. , 1090
SAYD, P. , 341
Sayd, P. , 42, 950
Schutte, K. , 439
Sebe, N. , 351
Senanayake, C. R. , 429
Shah, H. , 1110
Shaji, A. , 740
Shan, C. , 242, 469
Sharma, S. , 760
Siddiquie, B. , 740
Siebert, J. P. , 222
Skocaj, D. , 272
Skurikhin, A. N. , 202
Smith, C. , 1070
Smith, T. M. A. , 730
Smith, W. A. , 499
Song, Y.-Z. , 172
Soo, M. K. , 770
Sorri, I. , 252
Stein, A. N. , 2
Stenger, B. , 790
Su, H. , 419
Sucar, L. E. , 600
Sugaya, Y. , 282
Sun, L. , 32
Suter, D. , 740

Tannenbaum, A. , 22, 102, 322, 520, 890
Taylor, C. J. , 1060

Thacker, N. , 870, 990
Tiddeman, B. P. , 449
Tisse, C.-L. , 232
Tong, X. , 32
Torr, P. , 1120
Toussaint, M. , 12
Tresadern, P. A. , 800
Trucco, E. , 142
Trujillo, L. , 630
Tu, P. , 830
Twining, C. J. , 1060

Unnikrishnan, R. , 940
Uray, M. , 272
Uusitalo, H. , 252

van Gool, L. , 361
Vela, P. , 22
Vilarino, F. , 389
Vincent, G.-B. , 42
Vogiatzis, G. , 530
Voutilainen, R. , 252

Wallace, A. M. , 82, 750, 920
Wang, H. , 32, 331
Wang, Q. , 580
Wang, T. , 32
Wang, Y. , 620
Waydo, S. , 1000
Welch, G. F. , 720
Wheeler, F. , 830
Willert, V. , 12
Wilson, R. C. , 910
Wilson, R. G. , 381
Winter, M. , 459
Woodford, O. J. , 1120
Woodley, T. E. , 790
Wu, J. , 499

Xie, X. , 1040

Yan, B. , 770
Yan, F. , 650
Yan, W. , 580
Yang, B. , 32
Yang, S. , 32
Yeung, C. K. , 770
Yezzi, A. , 520
Yi, L. , 1020
Yu, J. , 162
Yu, X. , 1020

Zerubia, J. , 960
Zhang, Y. , 32
Zhijun, Z. , 770
Zhou, J. , 389
Zisserman, A. , 509, 570
Zivkovic, Z. , 810

Keyword Index

Active Contours, 409, 960, 1040
Active Vision, 62, 790, 890, 1040, 1090, 1100

Baysian/Graphical Networks, 12, 62, 590, 600, 680, 890, 910, 1120

Calibration, 232, 850, 1110
Clustering, 32, 62, 312, 890, 1000
Colour, 112, 820
Corner detection, 950, 1100

Data fusion, 690
Density estimation, 62, 72, 1110

Edge detection, 142, 399, 870, 1030
EM, 12, 32, 399, 530, 1000, 1060
Epipolar geometry, 192, 409, 439, 860, 950, 1110, 1120

Face and Gesture Recognition, 62, 449, 479, 489, 499, 800
Feature Extraction, 32, 62, 112, 222, 399, 479, 499, 800, 870, 1080
Fourier Transforms, 312, 660

Graph matching, 1030, 1080

Homographies, 192, 690, 850, 1100

Image Fusion, 222, 439, 690
Industrial Inspection, 142, 760, 820, 870
Invariants, 112

Kalman Filters, 52, 399, 1100

Level sets, 409, 900, 920, 960, 1040

Markov Models, 600, 700, 890, 910, 1080, 1120
Medical Image Analysis, 102, 302, 312, 489, 920, 1040, 1060, 1080
Model Based Vision, 142, 302, 489, 700, 760, 870, 960, 1080
Morphology, 32, 1060
Motion and Tracking, 12, 32, 52, 62, 72, 361, 399, 650, 680, 690, 740, 790, 820, 890, 920, 950, 1100, 1110
Multi-camera, 409, 690, 1120
Multi-media, 32, 112, 409, 489, 590, 600, 1030, 1120

Neural Networks, 1000

Object Recognition, 112, 142, 192, 212, 341, 399, 449, 479, 600, 870, 960, 1000, 1020, 1030, 1040, 1100
Optical Flow, 12, 102, 439

Particle filtering, 680
PDEs, 580, 900
Projective Geometry, 142, 361, 439, 690, 860, 1120

RANSAC, 52, 439, 850, 860, 950, 1050, 1100
Real-time vision, 52, 62, 72, 341, 399, 760, 790, 950, 1100
Registration, 102, 439, 489, 690, 850, 1060
Robotics, 52, 192, 232, 361, 760, 850, 860, 920, 950, 1030, 1090, 1100, 1110
Robust algorithms, 112, 302, 860, 950
ROC curves, 142, 192, 202, 479, 580, 1030

Satellite Image Analysis, 960, 1030
Scale space, 142, 222
Segmentation, 202, 222, 302, 409, 530, 820, 890, 920, 960, 1030, 1040, 1060, 1080

Shape analysis, 112, 312, 479, 499, 660, 760, 800, 870, 960, 1040, 1050, 1060
Statistical Shape/Appearance Models, 32, 302, 449, 479, 489, 499, 740, 790, 890, 1060
Statistics and Machine Learning, 12, 62, 312, 479, 499, 580, 590, 740, 1000, 1090
Stereo, 409, 910, 1120
Structure analysis, 700, 1050, 1060, 1080, 1100
SVMs, 479, 1090

Temporal Models, 12, 700
Texture, 112, 660, 700
Tracking People, 32, 62, 72, 680, 690, 740, 790, 800, 890, 920

View reconstruction, 52, 112, 409, 910, 950, 1120

Wavelets, 222, 960