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

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

Session 1: Tracking

Session 2: Surveillance/Registration

Poster Session 1

Session 3: Biometrics

Keynote Lecture

Session 4: Segmentation

VOLUME II

Session 5: Image/Video Retrieval

Poster Session 2

Session 6: Object Recognition

Session 7: Shape Modelling and Analysis

Session 8: Video Analysis

Topology-Preserved Diffusion Distance for Histogram Comparison

Wang Yan[†], Qiqi Wang[‡], Qingshan Liu[†], Hanqing Lu[†], and Songde Ma[†]
[†]National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences
{wyan, qsliu, luhq, masd}@nlpr.ia.ac.cn
[‡]Institute of Computational and Mathematical Engineering
Stanford University
qiqi@stanford.edu

Abstract

In most previous works, histograms are simply treated as n-dimensional arrays or even reshaped into vectors when measuring the distances between them. However many histograms have their intrinsic topologies, such as HSV histogram (cone), shape context (polar), orientation histogram (circle). The topologies are important for so-called cross-bin distance, because they determine the similarities between histogram bins, and influence the cross-bin distances between histograms. In this paper, we proposed the topology-preserved diffusion distance to take the topology into account. This method extracts the distance by measuring the heat diffusion process defined on the topology of the histogram. Moreover, a fast implementation with time complexity $O(N)$ is developed. Experiments on image retrieval and interest point matching show the effectiveness and efficiency of the proposed method.

1 Introduction

Histograms are widely used in many applications of image analysis and computer vision, such as interest point matching [8, 9], shape matching [2], image retrieval [12] and texture analysis [11]. They are very effective due to the rich information captured by the distribution. However, it is well known that histogram is sensitive to the changes of illumination and viewpoints, as well as quantization effects [2], therefore the design of a robust histogram distance is a challenging task.

According to the type of bin correspondence, histogram distance is divided into two categories [12], i.e. bin-to-bin and cross-bin distance. The former just compares each bin in one histogram to the corresponding bin in the other. The Minkowski distance (such as L_1 and L_2), histogram intersection, and χ^2 statistics belong to this category. These distances are sensitive to distortions, and suffer from the quantization effect. In contrast, the cross-bin distances allow the cross-bin comparison, and therefore are more robust to distortions. Quadratic Form distance (QF) [4], Earth Mover's Distance (EMD) [12], EMD-L_1 [7], EMD-Embedding [5], Pyramid Matching Kernel (PMK) [3] and diffusion distance [6] fall into this category.

Almost all of the previous works simply treated the histogram as an n-d interval. However in practice, many histograms have their special topological structures. For example, HSV colour histogram has a cone-shaped structure, orientation histogram is a circle, and shape context is based on the polar coordinate system. The simple treatment as an interval results in great distortions of the similarities between some bins, and then degrades the accuracy of the cross-bin distance. Take 1-d orientation histogram as an example. It's often represented as an interval $[0, 2\pi)$, though it's a circle actually. Given a small positive ε, two orientations 0 and $2\pi - \varepsilon$ are almost the same. However, with the traditional representation, the two locate at two extremes of the interval, respectively. The distance between them is almost the longest, which means the smallest similarity. It contradicts with human perception. The similar contradictions also exist in HS colour histogram with the first dimension for Hue and the second for Saturation, which is usually represented as a 2-d interval $[0,1) \times [0,1]$. Compared to the polar representation, the distances between colours locate at different sides of the line $H = 0$ are enlarged improperly, and the same for the distances between colours with small saturations. Similar problems exist in some other histograms, such as Scale-Invariant Feature Transform (SIFT) [8] and shape context [2], when they are represented as n-d intervals.

In the paper, we proposed the topology-preserved diffusion distance for histogram matching, which is inspired by Ling and Okada's work [6]. In their work, the cross-bin relations are simulated by the heat diffusion on the n-d interval, and the distance is the integral of the diffusion process. Different from [6], the proposed method solves the diffusion process on the histogram's intrinsic topology, rather than the interval. By preserving of the topology, it's more consistent with human perception. Sophisticated numerical method for Partial Differential Equation (PDE) is used to handle the non-trivial topology. Compared to the convolution in [6], it has solid mathematical background, such as the error bound and the numerical stability. The time complexity of the distance is $O(N)$, where N is the number of bins. The experiments are conducted on image retrieval and interest point matching. The proposed distance is compared with other state-of-the-art methods, and hypothesis tests are conducted to show its superior performance.

The rest of the paper is organized as follows. Section 2 discusses the related works. Our work is described in Section 3. Experiments are reported in Section 4 and then conclusion is drawn in Section 5.

2 Related Works

In this section, we briefly review the cross-bin distances, because our method belongs to this category. For more comprehensive discussion, please refer to [11, 12].

QF [4] is an early proposed cross-bin distance. Given two histograms h_1 and h_2, the distance is defined as

$$QF(h_1, h_2) = (h_1 - h_2)^T \mathbf{A} (h_1 - h_2), \tag{1}$$

where $\mathbf{A} = [a_{ij}]$ is the weight matrix and the weights a_{ij} denote similarities between bins i and j. In the comparison of colour histograms [4], the topology is taken into account by defining

$$a_{ij} = 1 - d_{ij}/d_{\max}, \tag{2}$$

where d_{ij} is the L_2 distance between colours i and j, and $d_{\max} = \max_{i,j}(d_{ij})$. QF makes each bin in one histogram to correspond to all the bins in the other, and thus tends to

overestimate the mutual similarity without a pronounced mode [12]. Different from QF, Our method use the diffusion process to simulate the cross-bin relations, and the bin in one histogram dynamically corresponds to some neighbouring bins in the other.

EMD dynamically selects the correspondences by solving a transportation problem. Although it achieves good performances in image retrieval [12] and texture analysis [11], its computation is costly, and usually large than $O(N^3)$, where N is the number of bins. Several fast approximations have been proposed. [5] embeds the EMD metric into a Euclidean space, and the EMD can be approximated by the L_1 distance in the space after embedding. Its time complexity is $O(Nd\log\Delta)$, where N is the number of features, d is the dimension of the feature space and Δ is the diameter of the union of the two feature sets. PMK [3] is proposed for feature set matching. First, a pyramid of histograms of a feature set is extracted, and then the similarity between two feature sets is defined by a weighted sum of histogram intersections at each level of the pyramid. EMD-L_1 [7] utilizes the special structure of the L_1 ground distances on histograms for a fast implementation of EMD.

The major difference between our method and the EMD related distances above is that the topology of the histogram is not considered in the latter. EMD uses ground distances defined on the n-d interval, and the other approximate methods are all developed for this specific type of ground distance. Although EMD may handle non-trivial topology by using properly defined ground distance, it's costly to compute ($> O(N^3)$). Our method is much faster ($O(N)$). Besides the major difference, our method differs from PMK in another two ways. First, PMK focuses on feature distributions in the image domain [3], while ours focuses on comparison of histogram-based descriptors, such as SIFT. Second, PMK uses intersection to allow partial matching, which is important for handling occlusions for feature set matching. In contrast, we employ the L_1 distance, because the histograms are all normalized.

Diffusion distance [6] measures histogram distance by heat diffusion. The difference of two histograms h_1 and h_2 is treated as the initial condition of a heat diffusion process $u(\mathbf{x},t)$, and the distance is defined as

$$K(h_1,h_2) = \int_0^T \|u(\mathbf{x},t)\|_1 \,\mathrm{d}t, \tag{3}$$

where T is a constant, and $\|\cdot\|_1$ represents the L_1 norm. [6] convolutes the initial condition with a Gaussian window iteratively to approximate the diffusion, and sums up the L_1 norms after each convolution to approximate the integral. The bin correspondences are implicitly determined by the diffusion. Its time complexity is $O(N)$, where N is the number of bins.

Similar to the diffusion distance, our method is also defined as the integral of the diffusion process. However, there are some significant differences. First, we define diffusion process on the histogram's intrinsic topological structure, while diffusion distance solves the process on an n-d interval. Second, we utilize numerical methods for PDE, i.e. finite volume method [1] and backward Euler scheme [10], to solve the diffusion process. In contrast, diffusion distance uses convolution to approximate the diffusion, which cannot handle the non-trivial topology.

3 Our Work

In this section, we first introduce the numerical method for heat diffusion equation, and then present the topology-preserved diffusion distance. At last, a fast implementation is described.

3.1 Numerical Method for Heat Diffusion Equation

We discretize the heat diffusion equation with Neumann boundary condition

$$\frac{\partial u(\mathbf{x},t)}{\partial t} = \nabla \cdot \nabla u(\mathbf{x},t), \quad \mathbf{x} \in \Omega, \tag{4}$$

$$\frac{\partial u(\mathbf{x},t)}{\partial \mathbf{x}} = 0, \quad \mathbf{x} \in \partial\Omega, \tag{5}$$

and then solve it numerically. The approach is briefly introduced as follows.

First, the spatial derivative $\nabla \cdot \nabla u(\mathbf{x},t)$ is discretized by finite volume method [1]. With division $\mathscr{D}$, the domain Ω is divided into N cells $\{c_k\}_{k=1}^N$, and the solution u is approximated in each cell as a constant, i.e.

$$u(\mathbf{x},t) \approx u_k(t), \quad \mathbf{x} \in c_k. \tag{6}$$

Integrating both sides of (4) over cell c_k, and using Gauss theorem and the boundary condition, we can approximate (4) and (5) with the spatial discretized equation

$$V_k \frac{\mathrm{d}u_k}{\mathrm{d}t} = \sum_{j \in \mathscr{N}_k} \alpha_{kj}(u_j - u_k), \tag{7}$$

where $\mathscr{N}_k$ is the set of neighbours of the cell c_k, and V_k and α_{kj} are constants related to the topology of domain Ω and the division $\mathscr{D}$ only.

By including the solutions of all cells, (7) can be rewritten in matrix form

$$\mathbf{M}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{A}\mathbf{u}, \tag{8}$$

where diagonal matrix $\mathbf{M}$ and operator matrix $\mathbf{A}$ consists of $\{V_k\}_{k=1}^N$and $\left\{a_{kj}\right\}_{k,j=1}^N$, respectively, and column vector $\mathbf{u} = [u_1, u_2, \ldots, u_N]^T$ consists of solutions in all cells.

Second, the time domain $[0,T]$ is discretized into a series of time steps $0 = t_0 < t_1 < \cdots < t_L = T$. Using the backward Euler scheme [10] to approximate the time derivative, the linear ordinary differential equation (8) becomes completely algebraic equation

$$\mathbf{M}\frac{\mathbf{u}^{(k)} - \mathbf{u}^{(k-1)}}{\Delta t_k} = \mathbf{A}\mathbf{u}^{(k)}, \quad k = 1, 2, \ldots, L, \tag{9}$$

where $\mathbf{u}^{(k)} = \mathbf{u}(t_k)$ is the solution at the k-th time step, and $\Delta t_k = t_k - t_{k-1}$. In numerical computation, we usually use fixed time step $\Delta t_k = \Delta t$. Defining matrix $\mathbf{B} = (\mathbf{M} - \Delta t\mathbf{A})^{-1}\mathbf{M}$, we can simply advance solution by

$$\mathbf{u}^{(k)} = \mathbf{B}\mathbf{u}^{(k-1)}. \tag{10}$$

Further more, we can get the solution at any time point directly by

$$\mathbf{u}^{(m)} = \mathbf{B}^m \mathbf{u}^{(0)}. \tag{11}$$

Due to the properties of the backward Euler scheme [10], our discretization (9) is stable for any positive time step Δt. The accuracies of both the spatial and temporal discretization are first-order. Therefore, the error in the numerical solution is $O(\Delta t) + O(\Delta \mathbf{x})$, where Δt is the size of the time step, and $\Delta \mathbf{x}$ is the size of the cells.

3.2 Topology-Preserved Diffusion Distance

Some notions are introduced first. A normalized histogram h is a probability density function defined on domain Ω, which is embedded in a normed space X. The topology of h is actually the topology of Ω. For example, the domain of colour histogram for Hue and Saturation is a disk embedded in the 2-d plane. The histogram $\hat{h}$ often referred in computer vision is the discrete version of h. It corresponds to a division $\mathscr{D}$, which divides Ω into cells $\{c_i\}_{i=1}^N$. The integral of h over a cell is the value of the corresponding bin in $\hat{h}$. We use "ˆ" to represent discrete histogram and other related functions.

To compute the topology-preserved diffusion distance between two histograms, the heat diffusion equation with their difference as the initial condition is solved first. And then, the distance is extracted by integrating the L_1 norm of the process along time. Given two histograms, $h_1(\mathbf{x})$ and $h_2(\mathbf{x})$, their corresponding initial condition is

$$u(0,\mathbf{x}) = h_1(\mathbf{x}) - h_2(\mathbf{x}). \tag{12}$$

Given the solution of heat diffusion equation (4) with conditions (5) and (12), the topology-preserved diffusion distance is defined as

$$K(h_1,h_2) = \int_0^T \int_\Omega |u(\mathbf{x},t)| \, \mathrm{d}\mathbf{x} \, \mathrm{d}t. \tag{13}$$

If Ω is an n-d interval and the division $\mathscr{D}$ is uniform, (13) reduces to the diffusion distance.

The method introduced in Section 3.1 is used to compare discrete histograms. Given two histograms $\hat{h}_1$ and $\hat{h}_2$, (4) and (5) are spatial discretized according to their common division $\mathscr{D}$, and the initial condition is

$$\mathbf{u}^{(0)} = \hat{h}_1 - \hat{h}_2. \tag{14}$$

We can get the discretized temperature field $\mathbf{u}(t)$ at any time t by (11). Since the integral over Ω can be approximated by L_1 norm, and the integral along time can be approximate by summation, (13) can be rewritten as

$$\hat{K}(\hat{h}_1,\hat{h}_2) = \sum_{i=0}^{L} \|\mathbf{u}(T_i)\|_1 \tag{15}$$

where $T_0 < T_1 < \ldots < T_L$ are time points. L is usually set to 2 or 3. The time complexity of this distance is $O(LN^2)$, where N is the number of bins. In the next section, a fast implementation is introduced, and its complexity is $O(LN)$.

A toy example is given in Figure 1 to illustrate the advantage of the proposed method. In the three Hue-Saturation histograms in Figure 1(a), only one bin in each is nonzero.

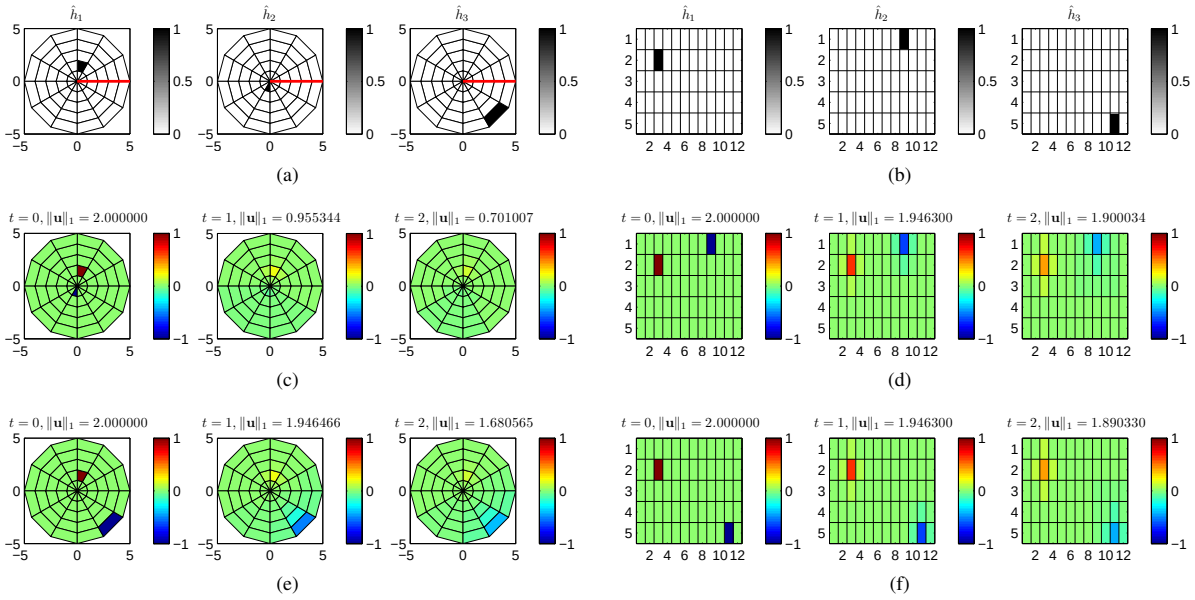


Figure 1: Toy example to show the advantage of the proposed method. (a) Histograms on disks. (b) Histograms on rectangles. (c) Diffusion process of $\hat{h}_1$ and $\hat{h}_2$ on the disk. (d) Diffusion process of $\hat{h}_1$ and $\hat{h}_2$ on the rectangle. (e) Diffusion process of $\hat{h}_1$ and $\hat{h}_3$ on the disk. (f) Diffusion process of $\hat{h}_1$ and $\hat{h}_3$ on the rectangle. Time points and L_1 norms of the temperature fields are shown above the images.

Intuitively, the similarity between $\hat{h}_1$ and $\hat{h}_2$ is larger than the one between $\hat{h}_1$ and $\hat{h}_3$, because the ground distance between the nonzero bins in the former pair is smaller. Cutting along the red line in Figure 1(a), i.e. $H = 0$, and performing some transformation, we get the common histograms in Figure 1(b). The diffusion processes on both disk and rectangle with different initial conditions are illustrated by Figure 1(c), (e), (d) and (f) respectively. The L_1 norms above the images show that the process in Figure 1(c) decays faster than the one in Figure 1(e). But there's no similar phenomenon in Figure 1(d) and (f). In fact, the L_1 norm of the last image in Figure 1(d) is even slightly larger than the corresponding one in Figure 1(f). The topology-preserved distances of Figure 1(c) and (e) are 3.6564 and 5.6270, respectively. This is consistent with the intuition. In contrast, the diffusion distances of Figure 1(d) and (f) are 3.2331 and 2.8826, respectively. Obviously, the diffusion distance fails in this case.

3.3 A Fast Implementation

Because of the linearity, the diffusion process with initial condition (12) can be viewed as the difference of two sub-processes, which use two histograms as the initial conditions respectively. The same holds in the discrete case. Plug (14) and (11) into (15), we get

$$\hat{K}(\hat{h}_1, \hat{h}_2) = \sum_{i=0}^{L} \left\| (\mathbf{B}^{m_i} \hat{h}_1) - (\mathbf{B}^{m_i} \hat{h}_2) \right\|_1, \tag{16}$$

where $m_i = \lfloor T_i/\Delta t \rfloor$. Since the division $\mathscr{D}$, the domain Ω, the time step Δt and the time points $T_0 < T_1 < \ldots < T_L$ are all predetermined, $\mathbf{B}$ can be computed in advance. Therefore both vectors, i.e. $\mathbf{B}^{m_i}\hat{h}_1$ and $\mathbf{B}^{m_i}\hat{h}_2$, can be computed at feature extraction step. The online computation only includes the differences of the vectors and the L_1 norms, and thus the online complexity is $O(LN) = O(N)$.

4 Experiments

The proposed methods are tested on natural image retrieval and interest point matching. Seven distances are compared, including L_1, L_2, χ^2, QF, EMD, Diffusion Distance (Diffusion) and Topology-Preserved Diffusion Distance (Topology). The weight matrix of QF is determined according to [4]. For the diffusion distance, we set $\sigma = 0.5$ as [6], and use 3×3 window for image retrieval and $3 \times 3 \times 3$ window for interest point matching. L_2 ground distance on the n-d interval is used in EMD. For the proposed method, we empirically choose time points $\{0,1,2\}$ for image retrieval and $\{0,2,4\}$ for interest point matching.

4.1 Natural Image Retrieval

This experiment is performed on the widely used Corel-5000 database [13], which consists of 5000 images. 8×8 HS colour histogram is used as the only feature. 1000 images (10 categories) with relatively significant colour characteristics are selected as the queries. For each query, the nearest 100 images are returned.

The average precisions of different distances are plotted in Figure 2 with respect to the scope. The time costs of different distances are shown in Table 1. EMD outperforms all the other methods, but its time cost is too high. The proposed method places the second, with much smaller time cost. L_1 and diffusion distance perform almost the same, and they are both the third. Although topology is taken into account, QF is worse than L_1, which is only a bin-to-bin distance. It confirms the analysis in Section 2, i.e. the static correspondence limits QF's performance. χ^2 and L_2 are the last.

Distance	Topology	Diffusion	L_1	χ^2	L_2	QF	EMD
Times (s)	18.0	14.1	6.3	13.4	7.2	238.4	8023.4

Table 1: Time costs in image retrieval

To further confirm the improvement, hypothesis tests are conducted. For a specific scope and a specific distance, the average precisions of 10 categories are treated as i.i.d. samples drawn from some distribution. The proposed method is compared with the others using these samples. Since the distribution is unknown, non-parametric Wilcoxon's signed rank test (one-sided) for two related samples is adopted. The p-values of the tests are listed in Table 2. Except EMD, all the others are small than 0.05, which means the improvements over the corresponding methods are all statistically significant.

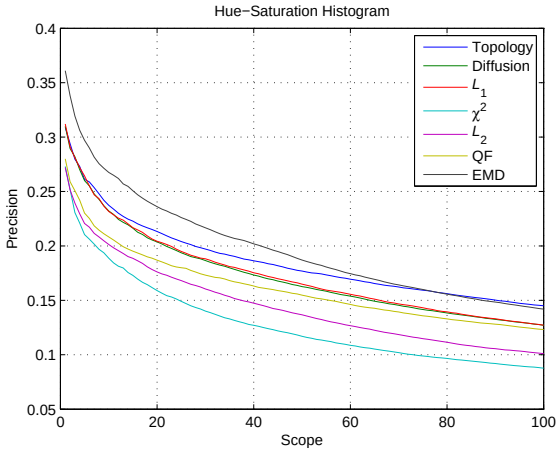


Figure 2: Retrieval precisions with respect to the scope in image retrieval

Scope	Diffusion	L_1	χ^2	L_2	QF	EMD
20	0.0469	0.0371	0.0039	0.0020	0.0020	0.5566
40	0.0020	0.0059	0.0039	0.0020	0.0020	0.6250
60	0.0039	0.0273	0.0039	0.0020	0.0137	0.7695
80	0.0098	0.0117	0.0059	0.0020	0.0420	0.6250
100	0.0039	0.0059	0.0039	0.0020	0.0322	0.6953

Table 2: p-values of hypothesis tests in image retrieval

4.2 Interest Point Matching

This experiment is performed on the Affine Covariant Regions Dataset [9], which consists of 40 image pairs with known plane projective transforms. We extract SIFT like descriptors from the interest regions detected by the Hessian-Affine detector [9]. The descriptor differs from SIFT by ignoring the tri-linear interpolation [8] and by being normalized by L_1 norm. The number of local descriptors varies from 200 to 4000 per image depending on the content.

The evaluation strategy in [9] is utilized. For each pair of images, the ground truth correspondences are first determined by the known transform. Then, we use the threshold-based strategy to match descriptors, i.e. two descriptors are matched if the distance between them is below a threshold. Varying the threshold, a Receiver Operating Characteristic (ROC) curve can be obtained. For some image pairs, it's hard to obtain the complete ROC curve with any distance because the precision keeps low. It's probably due to the limitations of the detector and/or the descriptor. For this reason, 21 image pairs are selected, and ROC curves in Figure 3 of different methods are the averages on these pairs.

Compared to image retrieval, similar ranking are shown in Figure 3. EMD is the best, followed by the topology-base diffusion distance. The diffusion distance and L_1 place the third, and then QF, L_2 and χ^2. The margin between Topology and Diffusion (or L_1) is

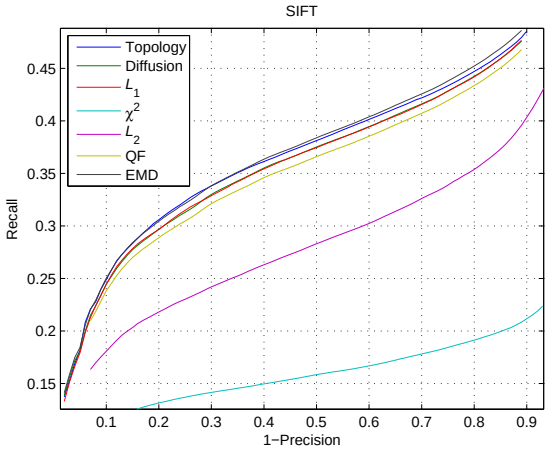


Figure 3: ROC curves in interest point matching

1−Precision	Diffusion	L_1	χ^2	L_2	QF	EMD
0.2	7.9802e-005	1.2267e-004	5.9570e-005	5.9570e-005	7.1872e-005	0.7823
0.4	0.0033	4.1887e-004	5.9570e-005	5.9570e-005	6.4356e-004	0.5829
0.6	6.1791e-004	5.4342e-004	5.9570e-005	5.9570e-005	3.5792e-005	0.8392
0.8	0.0037	4.1887e-004	5.9570e-005	5.9570e-005	5.0872e-005	0.5929

Table 3: p-values of hypothesis tests in interest point matching

roughly 1%. In spite of the superior performance, the computation of EMD costs about 300 hours. In contrast, our method uses only about 10 minutes, and the diffusion distance uses about 7 minutes.

The same hypothesis tests are conducted. For a specific precision and a specific distance, the recalls of different image pairs are treated as i.i.d. samples, on which the comparisons are based. The p-values are listed in Table 3. Again, the improvements over the other methods are significant, except EMD. Compared to Table 2, the p-values are smaller, which means the improvements are more significant in the sense of statistics, in spite of the smaller margins showed in Figure 3.

5 Conclusions

In this paper, we extend the diffusion distance by combining the idea of topology preserving. The proposed method defines the diffusion process on the topology of the histogram, and measures the distance by integrating the L_1-norm of the process along time. It outperforms most existing histogram distances by preserving the topology, and also outperforms topology-based QF by utilizing the diffusion process. Among the methods with complexities lower than $O(N^2)$, the proposed one is the most accurate. Moreover, it's also very efficient with the complexity $O(N)$.

Acknowledgement

This work is partially supported by the National Key Basic Research and Development Program (973) under Grant No. 2004CB318107, and the Natural Sciences Foundation of China under Grant No. 60405005, 60121302 and 60675003.

References

[1] T. Barth and M. Ohlberger. *Finite Volume Methods: Foundation and Analysis*, volume 1 of *Encyclopedia of Computational Mechanics*, chapter 15. John Wiley & Sons, West Sussex, 2004.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape context. *IEEE Trans. PAMI*, 24(24):509–522, 2002.

[3] K. Grauman and T. Darrell. The pyramid matching kernel: Discriminative classification with sets of image features. In *Proc. Int'l Conf. on Computer Vision*, 2005.

[4] J. Hafner, H. Sawhney, W. Equitz, M. Flickner, and W. Niblack. Efficient color histogram indexing for quadratic form distance function. *IEEE Trans. PAMI*, 17(7):729–736, 1995.

[5] P. Indyk and N. Thaper. Fast image retrieval via embeddings. In *Proc. Third Workshop on Statistical and Computational Theories of Vision*, 2003.

[6] H. Ling and K. Okada. Diffusion distance for histogram comparison. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2006.

[7] H. Ling and K. Okada. EMD-L_1: An efficient and robust algorithm for comparing histogram-based descriptors. In *Proc. European Conf. on Computer Vision*, 2006.

[8] D. Lowe. Distinctive image features from scale-invariant keypoints. *Int'l J. Computer Vision*, 60(2):91–110, 2004.

[9] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. PAMI*, 27(10):1615–1630, 2005.

[10] P. Moin. *Fundamentals of Engineering Numerical Analysis*. Cambridge University Press, 2001.

[11] Y. Rubner, J. Puzicha, C. Tomasi, and J. M. Buhmann. Empirical evaluation of dissimilarity measures for color and texture. *Computer Vision and Image Understanding*, 84(1):25–43, 2001.

[12] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *Int'l J. Computer Vision*, 40(2):99–121, 2000.

[13] H. Tong, J. He, M. Li, C. Zhang, and W. Ma. Graph based multi-modality learning. In *Proc. ACM Multimedia*, 2005.

Distance-Free Image Retrieval Based on Stochastic Diffusion over Bipartite Graphs

Christian Bauckhage
Deutsche Telekom Laboratories
10587 Berlin, Germany
http://www.telekom.de/laboratories

Abstract

We propose an approach to image retrieval that does not require any distance computations. The idea is to represent images and corresponding image features by means of the two sets of vertices of a bipartite graph. Even though in such a graph the images are not directly related, the degrees to which the features are present in an image allow for defining partial orders. If the degrees of presence are normalized such that they form probability distributions, similarity rankings result from the stationary distributions of stochastic diffusion processes over the graph. The method is closely related to recent approaches to ranking on manifolds but does not involve the computation of parameterized affinity and Laplacian matrices. Experiments with a standard image retrieval data set demonstrate the efficacy of the approach. Compared to a corresponding distance-based approach, it yields a higher overall precision.

1 Introduction

Content-based image retrieval (CBIR) from large databases has become a task of considerable practical importance. Admen, artists, designers, and journalists need fast access to appropriate icons or pictures to illustrate advertisements, journals, jingles or whatever else requires visual amelioration nowadays. However, the sheer size and speed of growth of present day image repositories create a crucial problem: consistent semantic annotations can hardly be provided single-handedly anymore. Neither can teamwork guarantee consistency. Experience with *folksonomies* gathered and maintained by online communities shows that spurious and ambiguous labels occur inevitably. Figure 1 illustrates what this implies in practice; it displays a choice from the 40 top ranking results obtained from typing "tiger" into Google's image search.

State of the art retrieval systems therefore apply computer vision techniques that are fine tuned to the task at hand by means of user feedback [3, 12, 14]. In the so called *human-in-the-loop* approach, the user repeatedly rates selections of images according to how well they match the current query. Based on this relevance feedback, characteristics of appropriate and inappropriate images are determined and a hopefully better suited set of images is retrieved from the database. This interactive process continues until the user's demands are met.

In a series of influential papers, Rui and Huang [11, 12] presented interactive CBIR systems based on a hierarchical model that combines different features and adaptable

Figure 1: Some of the top ranking results when searching Google Images for “tiger”.

distance measures. Even though their model and many of its successors enable flexible searches for images similar to the user’s intent, the way the different distances between features are defined appears to be solely technically motivated and is hard to grasp intuitively. More recent approaches [3, 14] apply more sophisticated reasoning and adaptation processes, but at their heart, too, lies the problem of defining distances between images that would allow for producing similarity rankings.

The reason why we emphasize this issue here is that it became clear some time ago that sets of images of a semantic class tend to form nonlinear manifolds whose global structure cannot be captured by simple metrics (see the examples in [2, 10, 13]).

Dealing with the problem of CBIR, the question then is how define similarities between objects residing on such manifolds. Or, in other words, what is needed is a method to rank such objects. As a matter of fact, this problem has been addressed in several recent contributions [1, 7, 8, 15, 16]. It has even been studied with respect to information retrieval in general [5] and image retrieval in particular [6]. Since these approaches are closely related to the idea presented in this paper, we will discuss them in more depth later on. For now, we simply point out that all these approaches derive the global structure of a set of data by considering local relations among individual elements which are again based on some notion of distance.

In this paper, we consider only a single iteration in an interactive CBIR system and focus on the problem of image ranking. Our approach determines similarities among images but does not require any distance computations. The idea is to represent a collection of images and a set of meaningful image features by means of the two sets of vertices of a bipartite graph. Assuming the edges between images and features to denote transitions in a Markov process immediately provides an ordering scheme: if we model a user query as an initial distribution over the vertices corresponding to images, a ranking results from the stationary distribution of a corresponding Markov chain that starts from this initial state.

In the next section, we detail this idea and the computational approach. We will see that there is a simple closed form solution to compute image rankings from an arbitrary query. We will discuss that, similar to the approaches in [1, 7, 8, 15, 16], our approach leads to a graph diffusion kernel. In contrast to existing methods, however, the kernel naturally results from the probabilistic model and its derivation does not require manual adjustment of free parameters. In section 3, we present experiments that demonstrate the efficacy of the proposed approach. On a standard data set it yields useful precision and outperforms a distance-based retrieval method considered for baseline comparison. Finally, section 4 concludes this paper and points out promising next steps of research.

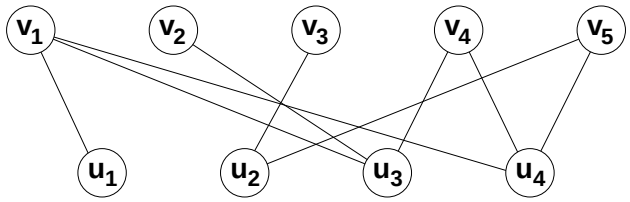


Figure 2: Example of a bipartite graph. Although there are no direct relations among the vertices u_i, their relations with the vertices v_j define a similarity and thus allow for partial ordering. With respect to vertex u_3, for instance, the order is $u_3 \sqsupseteq u_4 \sqsupseteq u_1 \sqsupseteq u_2$

2 Ranking as a Markov Process over Bipartite Graphs

The idea for the CBIR approach presented in this paper occurred while we were exploring novel mechanisms for collaborative filtering for automatic recommender systems. In the discussion that follows, we will thus frequently resort to rather metaphorical language and make use of terms such as *vote for* or *rate* which we feel convey the underlying ideas.

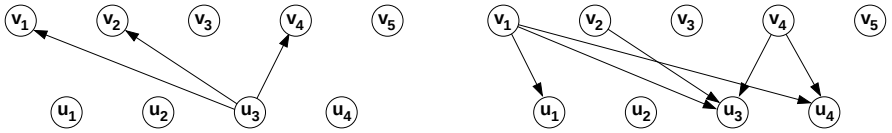
2.1 Mathematical Model

Assume labeled bipartite graph $G = (V,E)$ as shown in Fig. 2. Its sets of vertices V is partitioned such that $V = V_1 \cup V_2$ and $V_1 \cap V_2 = \emptyset$. The n vertices $u_1, u_2, \ldots, u_n$ in the set V_1 correspond to entities (such as users, images, ...). In a slight abuse of notation we will identify vertices and their labels and represent a labeling of the vertices in V_1 by means of a vector $\mathbf{u} = [u_1, u_2, \ldots, u_n]^T$. The m vertices $v_1, v_2, \ldots, v_m$ in the set V_2 correspond to rated items or features that are voted for (e.g. books, RGB color bins, gradient directions, ...) and their labels are stored in a vector $\mathbf{v} = [v_1, v_2, \ldots, v_m]^T$.

In a recommender systems, each entity $u_i \in V_1$ votes for (a subset of) the items in V_2. Dealing with CBIR, we may think of the votes as indicators to what extend a certain feature in V_2 is present in an image represented by u_i. In both cases, votes or frequency counts can be represented by means of directed, weighted edges (see Fig. 3(a)).

Even though there are no immediate relations (i.e. no edges) among the elements in V_1, their voting behavior allows for determining partial orders. Given an entity u_i, its fellow entities can be ranked according to how much their voting behavior resembles the one of u_i. In contrast to common distance measures between vectors of votes or frequency counts, the bipartite graph model seamlessly accounts for indirect relations as well. In the example shown in Fig. 2, for instance, u_2 is related to u_3 and u_4 alike. However, while the nature of its relation to u_4 is of first degree because both entities share a vote, its relation to u_3 is a second degree relation because it is mediated through u_4.

The key idea is now to understand relations of arbitrary degree as the outcome of a stochastic diffusion process over the bipartite graph. To this end, we normalize the votes cast by an entity so that they sum to 1. If all the votes of all the entities are stored in a column stochastic $m \times n$ matrix $\mathbf{R}$, and entity vectors are normalized so that they sum to 1, too, individual or weighted combined ratings result from $\mathbf{v}_t = \mathbf{R}\mathbf{u}_t$. With these assumptions, we obviously are considering probabilistic mappings from V_1 to V_2.



(a) $\mathbf{v}_0 = \mathbf{R}\mathbf{u}_0$ (b) $\mathbf{u}_1 = \mathbf{S}\mathbf{v}_0$

Figure 3: Example of the beginning of a stochastic diffusion process over a bipartite graph. Staring with the distribution $\mathbf{u_0} = [0010]^T$ produces a distribution $\mathbf{v}_0$ which in turn leads to the updated distribution $\mathbf{u}_1$.

Given the transition matrix $\mathbf{R}$, each item $v_j \in V_2$ *can deduce* which entities do vote for it for this information is essentially contained in the transpose of the transition matrix. If $\mathbf{S} \propto \mathbf{R}^T$ was normalized so that it is a column stochastic matrix, too, a set of rated items can (in turn) vote for entities (see Fig. 3(b)). An updated distribution over the entities in V_1 would then result from

$$\mathbf{u}_{t+1} = \mathbf{S}\mathbf{v}_t = \mathbf{S}\mathbf{R}\mathbf{u}_t \stackrel{!}{=} \mathbf{H}\mathbf{u}_t. \tag{1}$$

Note that the $n \times n$ matrix $\mathbf{H}$ introduced in last step of this derivation is a doubly stochastic matrix whose rows columns and rows sum to 1. It is square and non-negative and its eigenvalues λ_k are characterized by $|\lambda_k| \leq 1$.

Also, note that $\mathbf{H}$ defines a Markov process over the set V_1. Therefore, even though no direct relation among the $u_i \in V_1$ were available in the first place, we now have a tool for ranking. Assume an initial distribution $\mathbf{u}_0$ with only a few non zero entries. Then, after t steps, the probabilities in $\mathbf{u}_t = \mathbf{H}^t\mathbf{u}_0$ will be higher for entities which are more closely related to the initially active elements and less high for less closely related ones.

However, in this most simple form, the model cannot produce reasonable rankings if the underlying Markov chain is irreducible and contains positive-recurrent states. In this (practically very likely) case, the process converges to a uniform distribution over the elements in V_1 which does not allow for any ranking. We therefore assume the initial distribution $\mathbf{u}_0$ to be a steady source of probability mass that constantly feeds the stochastic process. With this modification, the update rule for distributions is given by

$$\mathbf{u}_{t+1} = \frac{1}{2}\Big[\mathbf{H}\mathbf{u}_t + \mathbf{u}_0\Big] \tag{2}$$

where the scaling factor $\frac{1}{2}$ ensures that $\mathbf{u}_{t+1}$ does sum to 1 just as $\mathbf{u}_t$ and $\mathbf{u}_0$ do. With some algebra it is easy to see that, written as a power series, the recursive expression in (2) amounts to

$$\mathbf{u}_t = \Big(\frac{1}{2}\mathbf{H}\Big)^t\mathbf{u}_0 + \frac{1}{2}\sum_{i=0}^{t-1}\Big(\frac{1}{2}\mathbf{H}\Big)^i\mathbf{u}_0 \tag{3}$$

Recall that $\mathbf{H}$ is a doubly stochastic matrix whose eigenvalues λ_k satisfy $|\lambda_k| \leq 1$. For the limit $t \to \infty$, we therefore have

$$\lim_{t\to\infty}\Big(\frac{1}{2}\mathbf{H}\Big)^t = 0 \quad \text{and} \quad \lim_{t\to\infty}\sum_{i=0}^{t-1}\Big(\frac{1}{2}\mathbf{H}\Big)^i = \Big[\mathbf{1} - \frac{1}{2}\mathbf{H}\Big]^{-1}. \tag{4}$$

Hence, the iteration in equation (2) is guaranteed to converge. Once the process has converged, the vector $\mathbf{u}$ it converged to is characterized by $\mathbf{u} = \frac{1}{2}\left[\mathbf{H}\mathbf{u} + \mathbf{u}_0\right]$ which directly leads to the closed form solution

$$\mathbf{u} = \frac{1}{2}\left[\mathbf{1} - \frac{1}{2}\mathbf{H}\right]^{-1}\mathbf{u}_0. \tag{5}$$

Therefore, given an arbitrary initial distribution $\mathbf{u}_0$ that might represent a single entity or –just as well– a mixture of entities, we can immediately determine the corresponding stationary distribution and the ranking it implies.

2.2 Discussion

It is interesting to note that the matrix in equation (5) constitutes a diffusion kernel [8]. In fact, from the derivation, we recognize another instance of the *kernel trick*. The similarities among vectors $\mathbf{u} \in \mathbb{R}^n$ that are contained in $\mathbf{H} = \mathbf{SR}$ result from mapping the vectors back and forth to a (usually higher dimensional) space $\mathbb{R}^m$.

Diffusion kernels for the purpose of computing similarities on manifolds or graphs have recently been studied by several authors [1, 7, 8, 15, 16]. In two contributions closely related to this paper, Zhou et al. [15, 16], investigate the problem of ranking on manifolds. They manifolds they are concerned with are represented by means of adjacency graphs. Given an unstructured set of feature vectors, they compute a matrix that represents local structures in the data by means of the distances between each data point and its k nearest neighbors. The adjacency matrix is then transformed into a similarity matrix $\mathbf{K}$ using a Gaussian kernel with parameter σ. Given $\mathbf{K}$, they show that diffusion processes on this adjacency graph are governed by the matrix $(1-\alpha)\left[\mathbf{1} - \alpha\mathbf{K}\right]^{-1}$. This, of course, closely resembles the result in (5).

In fact, from setting $\alpha = \frac{1}{2}$, we recognize stochastic diffusion over a bipartite graph to be a special case of the problem studied [15, 16]. However, some comments appear to be in order. While our derivation did not involve any free parameters, the approach by Zhou et al. requires at least three of them (k, σ, α). Moreover, while our approach avoids the computation of distances between vectors of ratings or features, the approach by Zhou et al. requires distance computation for constructing the adjacency matrix as well as the corresponding similarity matrix. Finally, the matrix $\mathbf{H}$ in our approach is a stochastic matrix and thus allows for a concise interpretation of the ranking procedure in terms of a Markov process. The matrix $\mathbf{K}$ in the approach by Zhou et al., in contrast, eludes such an interpretation.

Ranking on manifolds has already been applied in systems for document and image retrieval [5, 6]. However, to the best of our knowledge, all known such systems consider diffusion processes over adjacency graphs that represent local neighborhoods similar to the way discussed above. They therefore leave the user with the problem of choosing suitable distances and parameters. Our approach, on the other hand, is parameter-free. In the next section, we present initial experiments which demonstrate that it nevertheless yields useful results for the problem of CBIR.

3 Experiments

In this section, we report first results obtained from our approach to image similarity ranking. Note that, in our experiments, we did not pay too much attention to the selection of features suitable for the task of CBIR. Therefore, the figures and examples presented below should not be considered the maximum achievable performance. Rather, they are meant to illustrate the potential of ranking based on diffusion over bipartite graphs.

3.1 Setting

All our experiments considered the Corel 1000 data set [9]. It contains 1000 color images showing scenes or objects from 10 different categories; for each category, there are a 100 examples.

Since the idea of the *degree of presence* of a feature, which we alluded to in the last section, naturally translates to the use of histograms, we considered histogram-based descriptors to characterize entire images in the data set. In order to represent information due to the geometric structure of the image content, we decided to apply histograms of oriented gradients as introduced by Dalal and Triggs [4]. We used 12 bins to store gradient directions computed over a 9×9 grid of cells. The nonlinear normalization of different histograms was computed with respect to 3×3 blocks of cells. In order to represent information contained in the color distributions of the images, we adopted the idea by Dalal and Triggs to color histograms. Here, we considered 5×5 cells which again were normalized using 3×3 blocks. The color histograms in each of the cells contained 20 bins; the corresponding prototypical colors were determined from clustering the pixels of all images in the database into different sets. Other than that, no preprocessing steps were applied; in particular, we did not perform brightness adjustments or color normalization such as proposed in [4].

Given these image descriptors, we tested how our approach performed when the descriptors were considered individually as well as how it performed when they were combined into a larger vector. For baseline comparison, we also verified how a retrieval procedure performed that determines image similarities based on the cosine distance between feature vectors.

The figures in the Tables 1 to 3 resulted from issuing 10 different queries for each category and averaging over the results. In accordance with the traditional approach in information retrieval, we characterize the different algorithms in our test with respect to the *precision* they achieved.

3.2 Results

Tables 1, 2, and 3 list the precision values *at 5*, *at 10*, and *at 20*, respectively, and thus indicate how many relevant documents were returned among the top 5, top 10, and top 20 ranking documents. Results obtained from the histogram of gradients features are found in the columns marked *HOG*, the ones obtained from histograms of colors are displayed in the columns marked *HOC*, results yielded by the combined descriptors are labeled *both*.

Although some images seem to defy retrieval (e.g. the pictures of Mountains), the results obtained from stochastic diffusion processes over bipartite graphs generally appear reasonable and useful. Moreover, on average, our approach consistently outperforms the

	stochastic diffusion			cosine distance		
	HOG	HOC	both	HOG	HOC	both
New Guinea	**82**	**82**	**82**	10	50	44
Beaches	80	90	**94**	46	54	44
Rome	40	20	32	14	**44**	32
Busses	**80**	52	78	76	56	78
Dinosaurs	96	**100**	**100**	72	**100**	**100**
Elephants	20	**78**	70	22	66	50
Flowers	60	72	78	34	**94**	54
Horses	22	76	74	**96**	90	94
Mountains	4	24	18	8	**42**	34
Food	26	36	**66**	0	58	58
average	51	63	**69**	38	65	59

Table 1: Precision @ 5 obtained on the Corel 1000 data set.

	stochastic diffusion			cosine distance		
	HOG	HOC	both	HOG	HOC	both
New Guinea	**76**	**76**	**76**	5	52	41
Beaches	73	84	**90**	40	39	41
Rome	35	21	33	14	**37**	31
Busses	**74**	45	**74**	71	48	71
Dinosaurs	89	97	97	59	**100**	**100**
Elephants	21	**68**	65	15	61	42
Flowers	62	74	78	33	**88**	49
Horses	19	68	64	**92**	85	89
Mountains	4	21	17	5	**36**	25
Food	22	32	**59**	5	55	48
average	48	59	**65**	34	60	54

Table 2: Precision @ 10 obtained on the Corel 1000 data set.

baseline method, if it considers the combination of gradient and color features. Preliminary results like this are promising and justify further work on CBIR based on parameter-free diffusion over bipartite graphs.

Figures. 4 through 6 exemplify another interesting and promising feature of our approach: since it avoids the computation of distances, it does not only apply to ranking with respect to individual elements on a manifold but can be seamlessly applied in order to rank with respect to sets of elements. The figures illustrate, how this can aid CBIR.

In its lower row, Fig. 4 shows the top 5 ranking images that were returned when the image in the upper row was used as the query example. The ranking resulted from using the combined gradient and color features and starting the Markov chain with an initial

	stochastic diffusion			cosine distance		
	HOG	HOC	both	HOG	HOC	both
New Guinea	**65**	**65**	**65**	9	54	37
Beaches	69	77	**82**	32	38	40
Rome	**30**	18	26	15	27	26
Busses	66	37	**71**	60	48	64
Dinosaurs	80	90	90	42	**100**	98
Elephants	20	56	53	16	**57**	42
Flowers	58	66	74	27	**80**	50
Horses	21	58	60	**88**	79	86
Mountains	6	22	17	10	**35**	25
Food	19	34	**52**	7	**52**	40
average	43	52	**59**	30	57	51

Table 3: Precision @ 20 obtained on the Corel 1000 data set.

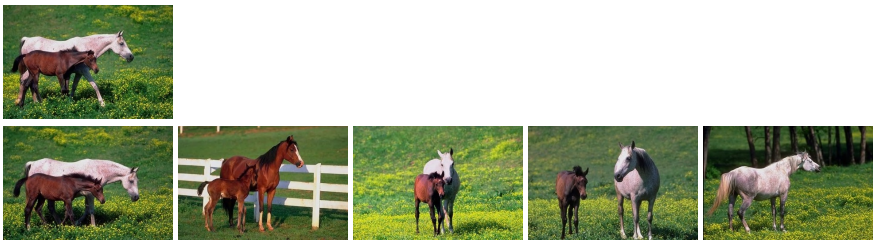


Figure 4: A single query image and the 5 top ranking results.

distribution $\mathbf{u}_0 = [0 \ldots 010 \ldots 0]^T$. Figures 5 and 6 show the outcome of the process when started with a distribution $\mathbf{u}_0 = \frac{1}{M}[0 \ldots 010 \ldots 010 \ldots 0]^T$ where $M = 3$ elements were set to $\frac{1}{M}$. From Fig. 5 we see that, if these elements index visually similar images, the retrieved images appear similar to these images, too. If the initial distribution covers a set of less similar images, the ones that will be returned among the top ranking images will also show a greater variety (see Fig. 6).

4 Summery and Outlook

In this paper, we described a novel approach to image ranking for content-based image retrieval. The interesting characteristics of this approach are that it is parameter-free and that it determines image similarities without computing distances. Given a collection of images together with a corresponding set of normalized feature vectors, the idea is to understand both sets as the disjoint sets of vertices of a bipartite graph. If the edges between images and features are assumed to denote transitions in a Markov process and if given queries are taken to be the initial distribution, an ordering with respect to a query results

Figure 5: Three similar query images and the 5 top ranking results.

Figure 6: Three less similar query images and the 5 top ranking results.

from the stationary state of the chain. By design –and in contrast to other recent approaches to manifold ranking– our approach allows for a rigorous interpretation in terms of Markov processes. Since these are completely characterized by the underlying stochastic matrix, a user does not have to adjust free parameters and distance measures. On the contrary, feature frequency counts or histograms immediately lead to necessary transition probabilities.

Preliminary results obtained with this approach are promising and justify further investigation as to what features might further improve precision. In addition, the method itself offers interesting perspectives for future research. An obvious idea is to apply it to classification: given a feature vector $\mathbf{v}$ derived from an unknown input image and a set of known images, the new image can be classified by, for instance, a majority count of the top ranking entities in the vector $\mathbf{u}$ that results from a query with the initial distribution $\mathbf{u}_0 = \mathbf{S}\mathbf{v}$. Another direction worth pursuing further appears from noting that equation (2) resembles the systems one deals with in linear quadratic control. The noticeable difference is that, in equation (2), the control matrix is set to $\mathbf{1}$. Especially from the point of view of interactive content-based retrieval, ways of adapting this matrix to better meet the user's intent seem a worthwhile topic.

References

[1] S. Agarwal. Ranking on Graph Data. In *Proc. ICML*, pages 25–32, 2006.

[2] M. Belkin and P. Niyogi. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, 15(6):1373–1396, 2003.

[3] I.J. Cox, M.L. Miller, T.P.Minka, T. Papathomas, and P.N. Yianilos. The Bayesian Image Retrieval System, PicHunter: Theory, Implementation and Psychophysical Experiments. *IEEE Trans. on Image Processing*, 9(1):20–37, 2000.

[4] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *Proc. CVPR*, volume 2, pages 886–893, 2005.

[5] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens. Random-Walk Computation of Similarities between Nodes of a Graph with Application to Collaborative Recommendation. *IEEE Trans. on Knowledge and Data Engineering*, 19(3):355–369, 2007.

[6] J. He, M. Li, H.-J. Zhang, H. Tong, and C. Zhang. Manifold-Ranking Based Image Retrieval. In *Proc. ACM Int. Conf. on Multimedia*, pages 9–16, 2004.

[7] H. Kashima, K. Tsuda, and A. Inokuchi. Kernels for graphs. In B. Schölkopf, K. Tsuda, and J.-P. Vert, editors, *Kernel Methods in Computational Biology*, pages 155–170. MIT Press, 2004.

[8] R.I. Kondor and J.D. Lafferty. Diffusion Kernels on Graphs and Other Discrete Input Spaces. In *Proc. ICML*, pages 315–322, 2002.

[9] J. Li and J.Z. Wang. Automatic Linguistic Indexing of Pictures by a Statistical Modeling Approach. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(9):1075–1088, 2003.

[10] S. Roweiss and L. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290(5500):2323–2326, 2000.

[11] Y. Rui and T. Huang. Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval. *IEEE Trans. on Circuits and Systems for Video Techology*, 8(5):644–665, 1998.

[12] Y. Rui, T. Huang, and S. Chang. Image Retrieval: Current Techniques, Promising Directions and Open Issues. *J. of Visual Communication and Image Representation*, 10(4):39–62, 1999.

[13] J.F. Tenenbaum, V. de Silva, and J.C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(550):2319–2323, 2000.

[14] J. Vogel and B. Schiele. Performance Evaluation and Optimization for Content-Based Image Retrieval. *Pattern Recognition*, 39(5):897–909, 2006.

[15] D. Zhou, O. Bousquet, T.N. Lal, J. Weston, and B. Schölkopf. Learning with Local and Global Consistency. In *Proc. NIPS*, number 16, pages 321–328, 2004.

[16] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf. Ranking on Data Manifolds. In *Proc. NIPS*, number 16, pages 169–176, 2004.

Word Co-occurrence and Markov Random Fields for Improving Automatic Image Annotation

H. Jair Escalante, Manuel Montes and L. Enrique Sucar
Computer Science Department
National Astrophysics, Optics and Electronics Institute
Puebla, 72840, México,
hugojair@ccc.inaoep.mx, {mmontesg,esucar}@inaoep.mx

Abstract

In this paper a novel approach for improving automatic image annotation methods is proposed. The approach is based on the fact that accuracy of current image annotation methods is low if we look at the most confident label only. Instead, accuracy is improved if we look for the correct label within the set of the top$-k$ candidate labels. We take advantage of this fact and propose a Markov random field (*MRF*) based on word co-occurrence information for the improvement of annotation systems. Through the *MRF* structure we take into account spatial dependencies between connected regions. As a result, we are considering *semantic* relationships between labels. We performed experiments with iterated conditional modes and simulated annealing as optimization strategies in a subset of the Corel benchmark collection. Experimental results of the proposed method together with a $k-$nearest neighbors classifier as our annotation method show important error reductions.

1 Introduction

The task of assigning semantic labels (words) to images is known as image annotation. This is a very important step towards developing more precise image retrieval systems. For text-based image retrieval systems, annotations are indispensable features; while for content-based image retrieval methods, annotations can provide them with semantic information for improving their performance. Image annotation, however, is not an easy task; manual annotation is both infeasible for large collections and subjective. Therefore, there is an increasing interest in developing automatic methods for image labeling.

There are two ways of facing this problem, at image level and at region level. In the first case, labels are assigned to the entire image as an unit, not specifying which words are related to which objects within the image. In the second approach, which can be conceived as an object recognition task, the assignment of labels is at region level; providing a one-to-one correspondence between words and regions. The last approach can provide more semantic information for the retrieval task, although it is more challenging than the former. Within the region-level automatic image annotation (*AIA*) task, we can distinguish two approaches for assigning labels to regions, these are soft and hard annotation. Hard

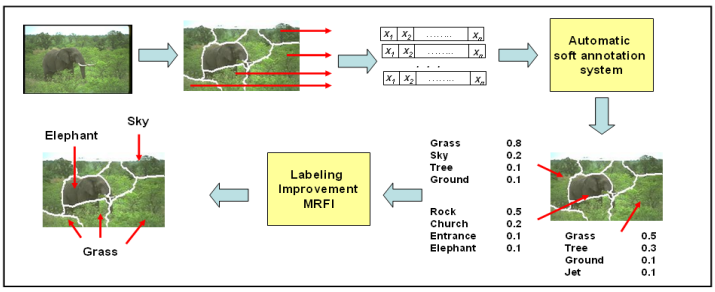


Figure 1: Graphical schema of our approach. We start from an image that is segmented into regions; attributes are obtained from each region; next these attributes are used with a soft-*AIA* method that returns a set of candidate labels, together with a relevance weight, for each region in the image. Then the method proposed in this paper is applied, and it returns an unique correct label for each image.

annotation consist of the task of assigning, with probability 1, an unique label to each region; soft annotation, on the other hand, ranks the labels according to their relevance to being the correct annotation for a given region. Accuracy of soft annotation systems is superior to that of hard systems, though assigning a set of labels to a single region is both confusing and impractical. On the other hand, accuracy of hard annotation systems is poor, though it is more understandable and practical assigning a unique label to each region.

In order to take advantage of the high precision of soft annotation methods as well as the clarity of hard approaches, we propose *MRFI*, a probabilistic model based on word co-occurrence information for improving image annotation systems. *MRFI* considers the top$-k$ candidate labels for each region within an image and, by using word co-occurrence information together with spatial context, it re-ranks each candidate label. Then we select the unique top label for each image, according to this ranking. In Figure 1 the proposed approach for improving *AIA* methods is graphically described. We used a $k-$nearest neighbor classifier as our *AIA* system and experiments on a subset of the benchmark Corel collection were performed. Experimental results show significant improvements by using *KNN+MRFI* over single *KNN*, furthermore *KNN+MRFI* outperforms several others state of the art annotation methods.

The rest of this document is organized as follows. In the next Section we review related work. In Section 3 some background information is described. Next in Section 4 the *MRFI* method is proposed. Then in Section 5 experimental results are presented. Finally, in Section 6 conclusions and future work directions are discussed.

2 Related work

A wide variety of methods for image labeling have been proposed since the late nineties. However, none of current methods have taken advantage of label's semantics for improving their performances. A very early attempt that used word co-occurrence information is the work by Mori et al [13], in which every word assigned to the entire image is inherited by each region; regions are visually clustered and probabilities of the clusters given

each word are calculated by counting the occurrence of common words within these clusters. A recent approach that attempts to take advantage of co-occurrence information is that proposed by Li et al [12]. They use a probabilistic support vector machine classifier for ranking candidate labels for each region within an image. Co-occurring words in the candidate labels for regions in the same image are weighted high; then candidate labels are re-ranked, top ranking labels are assigned as annotation for the entire image. Our approach is different to the previous methods because we obtained the co-occurrence information from an external corpus and considered spatial dependencies between connected regions. Instead of just considering co-occurrence of labels within the same image [12] or clusters of regions [13]. Moreover in such works co-occurrence information is used ad-hoc for their annotation method; while in this work we propose a method that can be used with other soft-annotation systems.

A work close in spirit to ours is due to Carbonetto et al [4]. In this work the authors introduce spatial information into a *MRF* for object recognition. This approach is different to the one we adopted; since Carbonetto et al define the potential function for discovering the unknown association between visual features extracted from each region and the considered labels; furthermore the *MRF* is entirely based on a single collection of annotated images. While in this work we use semantic information, obtained from an external source, for modeling word association between neighboring regions. Dealing with a different problem: that of selecting an unique label given a set (a subset of the vocabulary) of candidate ones; which can be seen as a re-ranking strategy. Conditional random fields (*CRF's*) have also been applied to pixel-level image labeling [9], and object recognition [14]. These works have obtained positive results in different scenarios, although their applicability is still limited to segmentation ([9]) and two-class object recognition ([14]). However using conditional random fields for *AIA* can be an immediate future work direction. The above described approaches take into account dependencies between connected regions [4, 9, 14]; although none of these have used semantic knowledge together with spatial context for improving performance of object recognition methods. *MRFI*, on the other hand, does not attempt to induce the *visual-features to word* relationship by considering spatial information. Instead *MRFI* takes advantage of semantic information and attempts to select the best configuration of labels for the regions contained in the same image. Semantic information is obtained off-line from a word co-occurrence matrix calculated from an external collection of manually annotated images.

3 Background

3.1 KNN as annotation system

The $k-$nearest neighbors (*KNN*) classifier is an instance based learning algorithm widely used in machine learning tasks. In this work we used this method as our annotation system due to the fact that it can outperform other state of the art methods (see Section 5); furthermore, *KNN* can be adapted to work in the hard and soft annotation schemas.

KNN starts from a training data set $\{X,Y\}$ consisting of N pairs of examples of the type $\{(x_1,y_1),\ldots,(x_N,y_N)\}$, with the $x_i's$ being $d-$dimensional feature vectors and the $y_i's$ being the labels of $x_i's$. In this work each x_i contains visual attributes extracted from a region. While each y_i is one of the $|V|$ labels we can assign to a region. The training phase of *KNN* consist of storing all available training instances. When a new instance, x_t,

needs to be classified *KNN* searches, in the training set, for $\{x_1^t, \ldots x_k^t\}$, the top $k-$objects more similar to x_t; then in a hard annotation schema it assigns to x_t the class of the most similar neighbor in the training set, we call this approach *1-NN*.

In order to apply *MRFI* with *KNN* as annotation method we need to turn *KNN* into a soft-annotation method. That is, candidate words for a given region should be ranked and weighted according to the relevance of the labels to being the correct annotation for such a region. We used the distance of the test instance to the top$-k$ nearest neighbors as relevance weight. In this way we can infer relevance weights directly related to the proximity of the neighbor to the test instance. Relevance weighting is obtained using Equation (1)

$$P^R(y_j^t) = \frac{d_j(x^t)}{\sum_i^k d_i(x^t)} \tag{1}$$

with $d_j(x^t)$ being the inverse of the Euclidean distance in the attribute space of instance x_j^t, within the $k-$nearest neighbors, to x^t, the test instance. As we can see, the sum of the priors for all the candidate labels is one, therefore this relevance weighting of *KNN* can be taken as the prior probability for the *MRFI* method. Note that this relevance weight is accumulative; that is, labels appearing more than once will accumulate their weights according to the times they appear in the top$-k$ labels. In this way we are implicitly accounting for repeated labels.

3.2 Obtaining co-occurrence information

Word co-occurrence is a form of word association that has been widely used by information retrieval models [1]. In the simpler schema, bags of words of documents and queries are compared (that is, word co-occurrences are calculated) for retrieving the documents whose bags of words are more *similar* to that of the query. This form of word association can be used with labels in the vocabulary for *AIA* tasks for taking into account semantic information between neighboring labels.

The co-occurrence information matrix (M_c) is a $|V|_X|V|$ square matrix in which each entry $M_c(w_i, w_j)$ indicates the number of documents (counted on an external corpus) in which words w_i and w_j appeared together. That is, we considered each pair of words $(w_i, w_j) \in V_X V$ and searched for occurrences, at document level, of (w_i, w_j). We did this for each of the $|V| * |V|$ pairs of words and for each document in our textual corpus. The collection of documents we considered for this work was the set of captions of a new image retrieval corpus: the *IAPR-TC12* [8] benchmark. This collection consists of around $20,000$ images that were manually annotated, at image level; therefore, if two words appear together in the captions of such collection, they are very likely to be visually related. Captions consist of a few text lines indicating visual and semantic content. From the entries of the M_c matrix we can estimate conditional and joint probabilities if we take: $P(w_i|w_j) = \frac{P(w_i,w_j)}{P(w_j)} \approx \frac{c(w_i,w_j)}{c(w_j)}$, and $P(w_i, w_j) \approx \frac{c(w_i,w_j)}{|D|}$, where $c(x,y)$ indicates the number of times x and y appear together in the corpus (that is, an entry of the M_c matrix); and $|D|$ is the number of documents in our textual corpus. If we repeat this process for each pair of words in the vocabulary we obtain a matrix of probabilities $((P_M))$, which may contain conditional or joint probabilities. Preliminary experiments showed that the use of conditional probabilities resulted in more significant improvements than those with joint probabilities; therefore, we used in this work conditional probabilities for (P_M).

A problem with the P_M matrix is the sparseness of data, that is, many entries of the matrix have zero values, which can affect the performance of our approach; this is a very common issue in natural language processing [6]. In order to alleviate this problem we applied a widely used smoothing technique known as interpolation smoothing [6], described on Equation (2)

$$P(w_i|w_j) \approx \Lambda * \frac{c(w_i,w_j)}{c(w_j)} + (1-\Lambda) * \frac{c(w_j)}{|W|} \tag{2}$$

where Λ is an interpolation parameter[1] and $|W|$ is the number of words in the collection. This formula is an interpolation between the empirical estimate ($\frac{c(w_i,w_j)}{c(w_j)}$) and the empirical distribution of the term w_j ($c(w_j)$). Therefore if two terms never co-occur in the co-occurrence matrix (M_c) we will not have a zero value in P_M.

4 *MRFI:* A Markov random field for improving *AIA*

A random field is a collection of random variables indexed by sites [11]. We consider a set of random variables $F = F_1, \ldots, F_M$ associated to each site in the site's system S. Each random variable takes a value f_i from a set of possible values L. A Markov random field (*MRF*) is a random field with the Markov property $P(f_i|f_{i-1}, f_{i-2}, \ldots f_1) = P(f_i|N(f_i))$, where $N(f_i)$ is the set of neighbors of f_i. A typical application of *MRF's* is to obtain the most probable configuration (F^*) for the *MRF*; given some restrictions represented by local probabilities, also known as potentials. We can express the joint probability of a *MRF*, $''F''$, given the observation, $''G''$, as the product of the potentials:

$$P_{F|G}(f) = \nu \prod_c P_c(X) \tag{3}$$

With ν constant, potentials ($P_c(X)$) can be thought of as restrictions that will favor or punish certain configurations of F. In this way, F^* can be considered as the configuration that have the highest compatibility with the local probabilities ($P_c(X)$). We can express the potentials as energy functions in exponential form, that is: $P_c(X) = e^{-U_c(X_c)}$, with $U_c(X_c)$ being an energy function. Then using Equation (3) we have an unique energy function $U_p(f) = \sum_c U_c(X_c)$. In consequence Equation (3) can be reformulated as:

$$P_{F|G}(f) = \frac{1}{Z} * \exp^{-U_p(f)} \tag{4}$$

with Z being a normalization constant. For a first order neighborhood, as the one we considered in this work, we have:

$$U_p(f) = \sum_c V_c(f) + \lambda \sum_o V_o(f) \tag{5}$$

Where V_c corresponds to P_F, the domain information given by the neighbors; and V_o corresponds to $P_{G|F}$, the information given by the observations; λ is a constant that weights the contribution of each term. In our case, we would like to select the best configuration of labels assigned to the regions in each image. Making a compromise between the visual

[1] Usually the value of Λ is chosen empirically. Intuitively a low value of Λ should be used with sparser data. After a few trial and error experiments we selected $\Lambda = 0.5$.

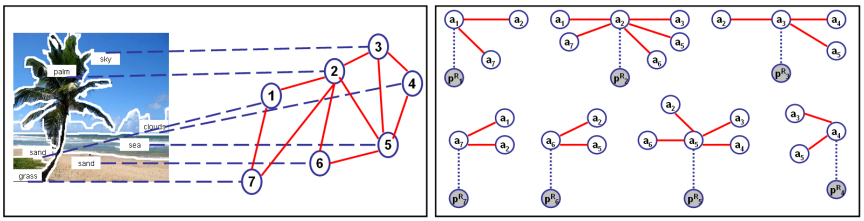


Figure 2: Left: graphical interpretation of *MRFI* for a given configuration of labels and regions. Right: spatial dependencies are shown for this configuration. The p_o^R's correspond to the relevance weight attached to each candidate label; the $a_i's$ represent the unknown association between connected regions.

properties of the region (V_o) and the semantics of its neighboring regions (V_c). Therefore, we used the above described framework for approaching this problem.

The observed variables in our task are the relevance weight attached to each label $p_1^R,\dots p_{M_n}^R$, for each region R; and the top$-k$ candidate labels $w_1,\dots w_K$, for each region. Observing this variables we define potential functions that exploit spatial dependencies between labels assigned to spatially connected regions within each image. The structure of *MRFI* and the dependencies it consider are shown in Figure 2. For this work we consider a region r_i is connected (spatially related) to another region r_j, if r_i is *next-to* r_j. Note that the next-to relation is symmetric and that *MRFI* depends on the segmentation. Moreover *MRFI* can not deal with problems like over-segmentation. However, as we will see in Section 5, if we have no available an accurate segmentation tool we can always divide an image into squared patches. Although poor, the use of this simple partition in *AIA* has outperformed methods based on sophisticated algorithms just has normalized cuts (see Section 5 and [4, 3]). Also we can make the square patches as small as we want; smaller patches will provide finer grain segmentations. Potentials for *MRFI* are defined in Equations (6) and (7) for the consideration of context and observation information, respectively.

$$V_c(f) = \sum_c (P(w_c|w_i))^n \tag{6}$$

$$V_o(f) = \left(\frac{1}{p_o^R(w_i)}\right)^n \tag{7}$$

Conditional probabilities in Equation (6) are obtained from the word co-occurrence matrix, as described in Section 3.2. While relevance weights p_o^R's, are obtained from the *AIA* system. The problem of selecting the correct annotation for each region within a given image reduces to the selection of the configuration that minimizes Equation (5). The selection of this *optimal* configuration is solved by standard optimization algorithms. In this work we performed experiments with two widely used algorithms: iterated conditional modes (*ICM* [2]) and simulated annealing with metropolis criteria (*SA* [10]). In Section 5 we report results of experiments with these two search strategies.

5 Experimental results

In order to evaluate the performance of *MRFI* several experiments on a subset of the Corel collection were performed. The data set we used is described in Table 1. It is a single

Data set	# Images	Words	Training blobs	Testing blobs
A-NCUTS	205	22	1280	728
A-P32	205	22	3288	1632

Table 1: Subset of the Corel image collection we used in the experimentation with *KNN-MRF*

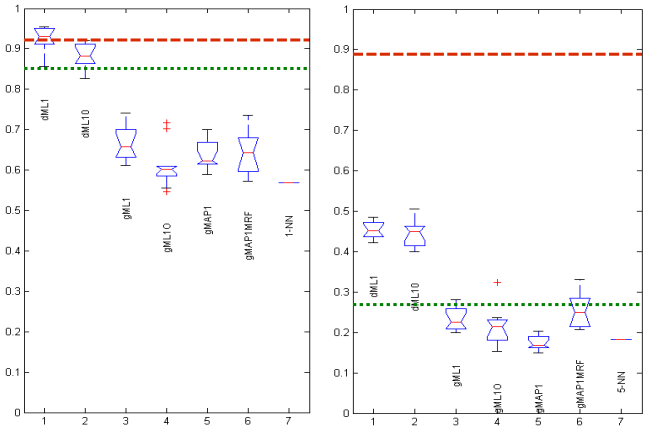


Figure 3: Comparison of *KNN* against other semi-supervised methods (*dML1* [7]; *dML1O*,*gML1*,*gMLO*, [3]; *gMAP1* [5]; *gMAP1MRF* [4]), using a Box-and-Whisker plot. The central box represents the values from the 25 to 75 percentile, outliers are shown as separate points. Left: accuracy at the first label. Right: accuracy at the top-5 labels. The upper dotted line represents a random bound, while the bottom dotted line represents a naïve method that always assigns the same label to all regions.

data set composed of 205 images segmented with normalized cuts [15] (*A-NCUTS*) and grid segmentation (*A-P32*). The attributes we considered for each region are the following: area, and color attributes. First we compared *KNN* against other semi-supervised object recognition methods [7, 4, 5, 3] (see caption of Figure 3), which are extensions and modifications to the reference work proposed by Duygulu et al [7]. In order to provide an objective comparison, we used the code provided by P. Carbonetto[2]. This code includes implementations of the above mentioned methods. In Figure 3 a comparison between *KNN* and the semi-supervised methods for the *A-NCUTS* data set is shown. In this plot, error is computed using the following equation:

$$e = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{M_n}\left(1-\delta(a_{nu}^{-}=a_{nu}^{max})\right) \tag{8}$$

where M_n is the number of regions on image n, N is the number or images in the collection; and δ is an error function which is 1 if the predicted annotation a_{nu}^{max} is the same as the true label a_{nu}^{-}. Results with the test sets are averaged over 10 trials.The left plot in Figure 3 shows error at the first label (*hard annotation*). Error is high for all of the methods we considered, however *1-NN* outperforms in average all of the semi-supervised approaches.

[2] $http://www.cs.ubc.ca/\sim pcarbo/$

Method	**k**	**Its**	λ	**n**	**Context**	*Time*	**Improved**	#−**runs**	**AVG-I**
ICM-P32	20	100	0.1	1	Next-to	1.8	134	4500	41.3
ICM-NCUTS	20	100	5	0.5	Full	0.78	56	4500	-0.7
SA-P32	20	50	0.1	2	Next-to	1.5	144	2700	98.6
SA-NCUTS	20	25	10	0.5	Next-to	0.5	54	2700	27.5

Table 2: Parameters for the best configurations. k is the number of candidate labels in *KNN*; *Its* is for iterations; λ and n are parameters for Equation (5); context indicates the type of neighborhood considered; time is the average time in seconds required to analyze an image with *MRFI*. *Improved* is the number of annotations improved. #−*runs* is the number of experiments performed and *AVG-I* is the total of annotation improvements averaged by #−runs

gMl0 is the closest in accuracy to *1-NN*, though it obtains an average error which is above *1-NN* by 4.5%. In the right plot of Figure 3 we consider a label is correctly annotated if the true label is within the top−5 candidate labels, (*soft annotation*). As we can see, error for all methods is reduced, this clearly illustrates the fact that accuracy of annotation systems is high considering a set of candidate labels instead of the first one. In this case *gMAP* [5] outperforms *5-NN* by 0.9% in average. All other approaches obtain a higher average error than that of *5-NN*.

In the second experiment we compared the performance of *KNN+MRFI* to that of *KNN* alone as well as to the previous methods. Note that we have several parameters to fix for *MRFI*. These are: k, the number of candidate labels for each region; λ and n, parameters for Equation (5); the number of iterations is a parameter for the optimization algorithms; furthermore, we performed experiments with spatial context (see Figure 2) and with full spatial context, that is, assuming all regions in an image are connected to each other. Given that *MRFI* is an efficient method we could perform many experiments with both data sets in order to determine the average improvement of *MRFI+KNN* over single *KNN*. The parameters of the best configurations for each data set considering both optimization strategies are shown in Table 2. We also show the average of accuracy improvement and processing time. From Table 2 we can point out several interesting observations. First, as expected, the more candidate labels we consider, the more improvements we gain. We performed experiments with $k \in \{3,5,10,20\}$ and the best results were obtained with $k = 20$. *ICM* needs a higher number of iterations to converge than *SA*. A small value of λ works well for the *P32* data set, which means that a small weight is given to the co-occurrence information. While a high value of λ performs better for *NCUTS*, giving more importance to co-occurrence information. We can see that for *NCUTS* a value of $n = 0.5$ performs well, while this parameter do not significantly affected the performance of *MRFI*. The use of spatial information, through the *next-to* relation, results in larger improvements than if consider each region is connected to each other in the image. Improvements are consistent through the number of experiments performed. The lowest average improvement was obtained with *ICM-NCUTS*. While with the grid segmented data (*P32*) we obtained the largest improvement, 98 annotations per run in average; which is a very significant improvement. An important result showed in Table 2 is the processing time[3] required to process an entire image with *MRFI*. These results show the efficiency of *MRFI*.

[3] All experiments were carried out on a PC with 1 GB in RAM and a 2.7 GHz *pentium*^{R} processor

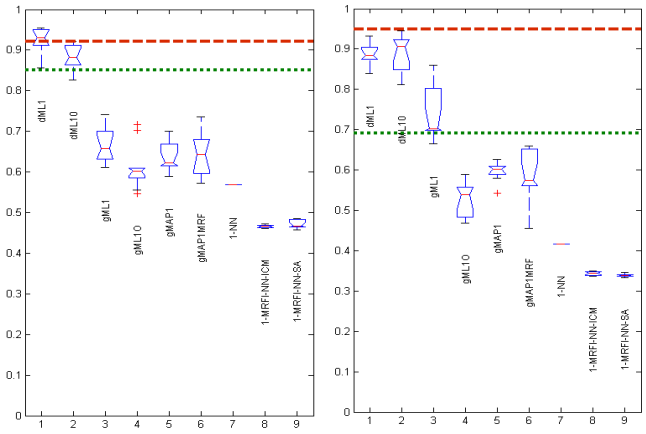


Figure 4: Comparison of *KNN* and *KNN+MRFI* against other semi-supervised methods (see caption of Figure 3) for images segmented with normalized cuts (left) [15] and with the grid approach (right); error is measured at the first label, see caption in Figure 3.

In all experiments performed using grid segmentation, which is faster than the other method, outperformed in accuracy segmentation with normalized cuts [15]. This result agrees with previous work [4, 5]. In *MRFI* this can be due to the fact that with grid segmentation (*P32*) the structure of the *MRF* is equal for all images. While for normalized cuts we have a different segmentation, according to the image's content, and therefore a different structure for the *MRF*. The use of *SA* instead of *ICM* does not result in significant improvements, *SA* outperformed *ICM* by 0.5%, which means that we have not many local minima. In Figure 4 we compare the best configurations of *MRFI* (Table 2) with the other methods. From Figure 4 we can clearly appreciate the improvement we can get by applying *MRFI+KNN*, instead of *1-NN* alone, for both data sets. The improvements of *MRFI+KNN* over *1-NN* are of 7.5% and 10.3% for the *P32* and *NCUTS* data sets, respectively. These percentages represent around 140 (for *P32*) and 46 (for *NCUTS*) annotations that were enhanced; this is a very significant improvement in accuracy. Furthermore, the difference in performance between *MRFI+KNN* and the other methods is dramatically increased. The semi-supervised method with closest average accuracy is *gML0*. *MRFI+KNN* improved *gML0* in average by 18.9% and 14.7% for the *P32* and *NCUTS* data sets, respectively. Results from this Section give evidence that *KNN+MRFI* is an effective image annotation method. Furthermore, *MRFI* can be applied with any other annotation system, though more experimentation should be performed in order to evaluate its impact with other methods.

6 Conclusions

We have presented *MRFI*, a method for the improvement of *AIA* systems. In *MRFI* spatial dependencies are considered through a *MRF* model. Semantic information between labels is incorporated using word co-occurrences. Co-occurrence information is calculated off-line from an external collection of captions, which is a novel approach. Experimen-

tal results of our method on a subset of the Corel collection, give evidence that the use of *KNN+MRFI* results in significant error reductions. Our method is efficient since the co-occurrence matrix is obtained off-line, and in most of the cases we just need a few iterations to obtain a good configuration (around 1.1 seconds per image). Furthermore, *MRFI* can be used with other *soft-annotation* systems.

The improvement of the co-occurrence matrix is an immediate step towards the enhancement of *MRFI*. Other future directions include the consideration of global image labels into *MRFI* and considering other models than *MRF's*, such as *CRF's* as well as experiments with probabilistic *AIA* methods.

Acknowledgements. We would like to thank K. Barnard, P. Carbonetto and M. Grubinger for making available their data and the reviewers by their useful commentaries that helped to improve this paper. This work was partially supported by CONACyT under grant 205834.

References

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Pearson E. L., 1999.

[2] J. Besag. On the statistical analysis of dirty pictures. *J. Roy. Stat. Soc. B*, 48:259–302, 1986.

[3] P. Carbonetto. Unsupervised statistical models for general object recognition. Master's thesis, C.S. Department, University of British Columbia, August 2003.

[4] P. Carbonetto, N. de Freitas, and K. Barnard. A statistical model for general context object recognition. In *Proc. of 8th ECCV*, pages 350–362, 2005.

[5] P. Carbonetto, N. de Freitas, P. Gustafson, and N. Thompson. Bayesian feature eeighting for unsupervised learning. In *Proc. of the HLT-NAACL workshop on Learning word meaning from non-linguistic data*, pages 54–61, Morristown, NJ, USA, 2003.

[6] S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. In *Proc. of the 34th meeting on Association for Computational Linguistics*, pages 310–318, Morristown, NJ, USA, 1996.

[7] P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Proc. 7th ECCV*, volume IV of *LNCS*, pages 97–112. Springer, 2002.

[8] M. Grubinger, P. Clough, and C. Leung. The iapr tc-12 benchmark -a new evaluation resource for visual information systems. In *Proc. of the International Workshop OntoImage'2006 Language Resources for CBIR*, 2006.

[9] X. He, R. Zemel, and M. Carreira. Multiscale conditional random fields for image labeling. In *Proc. of CVPR'04*, volume 2, pages 695–702. IEEE, 2004.

[10] S. Kirkpatrick, C. Gelatt, and M. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[11] Stan Z. Li. *Markov Random Field Modeling in Image Analysis*. Springer, 2nd edition, 2001.

[12] W. Li and M. Sun. Automatic image annotation based on wordnet and hierarchical ensembles. In *CICLING*, volume 3878 of *LNCS*, pages 417–428, Mexico, City, 2006.

[13] Y. Mori, H. Takahashi, and R. Oka. Image-to-word transformation based on dividing and vector quantizing images with words. In *1st Int. Worksh. on Multimedia Intelligent Storage and Retrieval Management*, 1999.

[14] A. Quattoni, M. Collins, and T. Darrel. Conditional random fields for object recognition. In *NIPS*, 2004.

[15] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI-IEEE*, 22(8):888–905, 2000.

Image Retrieval through Qualitative Representations over Semantic Features

Zia Ul-Qayyum, A.G. Cohn
zia@comp.leeds.ac.uk, A.G.Cohn@leeds.ac.uk

Abstract

We propose a qualitative knowledge-driven semantic modelling approach for image retrieval based on qualitative relations over local semantic concepts of images. The relative similarity of two images is proportional to their qualitative similarity. The similarity measure is calculated for each query by exploiting the notion of conceptual neighbourhood – a measure of closeness between qualitative relations. The approach is motivated by the need to perform semantic querying using qualitative relations and bridge the semantic gap between a human user and that of CBIR systems. Three qualitative representations (and several variants) and a corpus of 700 natural scene images have been used to evaluate the effectiveness of image retrieval using this approach.

1. Introduction

Advances in digital technologies along with the growth of the Web have resulted in universal access to very large archives of digital data. This has lead to an increasing requirement for systems with more flexible and robust techniques to handle dynamic and complex visual content at a higher semantic level. Content based image classification and retrieval systems have thus gained more importance and have become an active research area [1]. In all such systems, image interpretation and understanding plays a vital role. Most of the research in this area is primarily based on use of low level image features like colour, texture, shape etc [9, 18]. Although low level image processing algorithms and methodologies are quite mature, such systems are hard to be used effectively by a novice due to the semantic gap between user perception and understanding, and system requirements. Bridging this gap between low level synthetic features and high level semantic meanings is, therefore, generally regarded as an open problem [1]. Humans tend to describe scenes using natural language semantic keywords/concepts like sky, water etc and specify queries like “an image with water next to fields and sky above....” or “… has a small lake with high peaks of mountains behind and fields on left...”. This suggests that use of underlying semantic knowledge in a qualitative representation language may provide a way to model the human context and is a natural way to bridge semantic gap for better image understanding, categorization and retrieval capabilities.

This paper thus proposes a qualitative knowledge-driven semantic modelling approach for IR. Qualitative representation of the local semantic contents of an image allows for representation and reasoning of content structures at a higher abstraction level than low level features. In earlier work [13], we showed how category descriptions for a set of images could be learned using qualitative spatial representations (QSR) over a set of local semantic concepts (LSC) such as sky, grass. There were six global categories (e.g. coasts, forest etc) [19] and we used three kinds of QSR techniques to demonstrate that supervised learning using QSR of semantic image concepts can rival a non qualitative approach for image categorization [19,13], and moreover result in a more intuitive and more human understandable image description.

Our hypothesis in this paper is that the qualitative representations which were able to effectively support categorization may also provide an effective and natural way to support content-oriented querying. A query can either be directly described in the qualitative representation, or in the evaluation of our approach described below, a query can be given as a sample image (i.e. query by example: QBE) – the system then forms a qualitative description of it by a conjunction of qualitative relations between the semantic concepts. In both cases the system then compares the query qualitative description with qualitative descriptions of images in the database of images, and uses a *qualitative similarity measure* to retrieve qualitatively similar images, and show how retrieved images can be ordered accordingly. We do not assume that images have already been assigned categories/classes. The qualitative similarity measure is based on the notion of a *conceptual neighbourhood* (CN) [10] – see §4.

In experiments, using this technique on the different QSRs, we observed that the various representations had different levels of performance for different categories of images; this lead us to investigate the use of voting schemes in order to combine the different QSR to enhance the performance of the retrieval system overall.

A quantitative metric based evaluation of approaches based on qualitative representations has always been difficult. In order to evaluate the performance of this approach to IR, we take advantage of manually assigned categories for the image DB in our experiments. Although we are not performing image categorization, and the retrieval algorithm does not use the category information, success of retrieval is evaluated by counting the number of highly ranked images in the same category as the query.

The experimental data set is a collection of 700 natural scenes images, provided and hand labelled with categories by Vogel et al, who developed a semantic modelling framework for image categorisation and retrieval [19]. Our approach builds on her work, an overview of which is presented in §3.

The rest of the paper is structured as follows. Related work is briefly discussed in §2. §3 describes our approach to image description using QSR. A qualitative similarity based IR approach is presented in §4. §5 presents the results and evaluation of the approach, while §6 presents our conclusions and suggestions for future work.

2. Related Work

In the IR literature, image description and better understanding of underlying semantic content play important roles as the nature and structure of the query depends on the underlying image description. In this section, we first we describe the most relevant work from allied disciplines of content-based IR and then briefly survey the field of QSR.

CBIR systems have become an active research area in computer vision. [7,9,15,18] review the state of the art in segmentation, indexing and retrieval techniques in a number of CBIR systems. Despite increased work in aspects related to high level semantics of image features, the gap between low level image features and high level semantic expressions is a bottleneck in accessing multimedia data from databases. These surveys reveal that almost all existing approaches rely on using low level image features for image description, categorization and retrieval. Since image understanding is key to all content-based image categorisation and retrieval systems, so a human understandable image description may yield more robust systems since humans normally tend to use semantic and qualitative terms to describe a situation/image. Therefore, a retrieval system based on qualitative description of underlying semantic knowledge may help a non-expert user query such systems more effectively. Research has already been done focusing on the use of labelling the image regions with semantic concepts and carrying out key-word based IR. One such probabilistic approach [4] is to assign small image areas labels such as “man-made” and “natural”, and global labels such as “inside”, “outside” to whole images using class likelihoods from colour-texture features of images for semantic IR. Local regions of images have been annotated with 11 and 10 semantic

categories respectively [17,20]; in [17] a global label is not assigned to images, so retrieval is based on local semantic concepts only. An IR approach based on semantically labelled image regions is demonstrated in [1]. These image regions have been hierarchically classified based on their semantics using low level image features. Retrieval is based on these semantic keywords attached to particular images.

In an approach [21] for semantic retrieval based on content and context of image regions and which supports both keyword and QBE queries, images are segmented using a semantic codebook based on colour and texture classification. The content and context describe a region's low level features and their relationships respectively. It uses only dominant semantic categories of an image and the most typical images in that category are selected manually from an image database which can best model the codebook representing colour and texture classification for that particular semantic category. Another query by semantic example (QBSE) approach is based on posterior concept probabilities of each concept in an image [14]. QBSE is accomplished by comparing the probability simplexes of the query image and all database images to find the closest neighbours. The perceptual segmentation approach in [8] has not been applied in their work for image categorization and retrieval, but the relative effectiveness of their approach to image segmentation and labelling can be used to perform keyword based IR. The VISENGINE system [16] relies on segmenting image regions by clustering visual features like colour, texture, shape etc and differentiating them into foreground and background regions. The approach is largely user-centred, and therefore results may vary depending on human perception and context. Since only large regions are identified during segmentation, small image areas do not contribute towards the retrieval process which may inhibit a true semantic similarity in the retrieved images. Progress can also be made algorithmically, e.g. it has been shown that classification and retrieval accuracy can be boosted by combining different approaches [11]. The use of ontologies and metadata representation languages is another recent trend for annotating and retrieving images [12]. A prerequisite for this approach is the construction of generic and possibly domain specific ontologies from which the detailed annotations are constructed.

One crucial research question for QBE systems is how to measure the level of similarity, and assess the accuracy of such a technique. Defining a notion of similarity is difficult since context may play a pivotal role. Moreover, when using a qualitative representation, where feature descriptions do not take quantitative values, the very notion of a metric becomes problematic; approaches to qualitative similarity are discussed in [3]. In computer vision and image processing, metric approaches have generally been used to compute scene similarity, e.g. a measure based on normalised distance for a semantic ordering of natural scenes in categories such as forest and mountains, mountains and rivers/lakes [19].

The field of QSR has become increasingly more active within AI as it arguably provides cognitively or intuitively relevant representations for spatial information – typical spatial expressions in natural language are qualitative rather than quantitative. Moreover, qualitative representations abstract away from noise and uncertainty in perceptual data. It has increasingly been used in different application domains like GIS, NLP, robotics, computer vision etc, see [6] for a review. There are many QSR, covering aspects such as topology, distance, orientation, and shape. Rather than attempt an exhaustive analysis of the utility of all these calculi, we concentrate on a small set of QSR here; we do not claim these are necessarily the best calculi for image description, or even for the particular kinds of images in the database we use here, but leave that for further work. Our aim is simply to illustrate the use of qualitative calculi for IR and to demonstrate their potential applicability and suitability for CBIR.

In the qualitative framework, in which images are described using a small finite set of relations or qualitative values, similarity can be computed by using the distance in the CN graph. The notion of a CN was first put forward [10] in the context of a set of 13 pairwise and disjoint relations between temporal intervals and was defined as "two spatial

or temporal relations are conceptual neighbours if one can be transformed into the other by a single [continuous] transformation/transition". Given two such qualitative image descriptions, their similarity is proportional to the number of such transformations required to turn one into the other [5].

3. Qualitative Image Description

Our approach builds on Vogel et al's work [19] in which images from a 700 image corpus were divided into a grid of 10x10 regions (instead of using segmentation techniques) and nine local[1] and discriminating semantic concepts were identified: sky, water, grass, foliage, flowers, field, mountain, snow, trunks and sand. Vogel et al manually annotated 99.5% of the images with these concepts, and used this as input to supervised learning techniques to annotate image patches automatically. A label "rest" is used for unidentified patches or occurrences of other semantic categories. Images were represented by frequency histograms of local semantic concepts and based on a semantic typicality measure; images were categorized into one of the six semantically meaningful categories sky_clouds (34), coasts (143), landscapes_with_mountains (lwm) (178), fields (128), forests (103), waterscapes (114). (The numbers in brackets show total number of images for the respective category.) This approach is partially spatial through its division of the image into horizontal bands (e.g. top (T), middle (M) and bottom (B)) but is mainly based on the metric value of the percentages of discriminant semantic concepts.

We use the hand labelled data set in the experiments reported here in order to evaluate using the "gold standard" rather than be affected by the particular model learned for annotation. The images are described using the following QSRs:

1) The relative size (measured in grid squares) for all possible pairwise combinations of the semantic labels. Each may be regarded as an attribute of the image with possible values of 'Greater than' (>), 'Less than' (<) and 'Approximately Equal to' (≈) – we allow a ±10% tolerance for ≈.

2) Allen relations [2] (measured on vertical axis between the intervals representing the maximum vertical extent of each concept occurrence). The 13 relations are: 'before' (<), 'meets' (m), 'overlaps' (o), 'during' (d), 'starts' (s) and their inverses 'after' (>), 'met-by' (mi), 'overlapped-by' (oi), 'contains' (di), 'started-by' (si), 'finished-by' (fi) respectively, and 'equal' (=). A 14th relation 'no' is used if neither attribute is present.

3) Chord patterns [15] of semantic concepts applied to each grid row. Each semantic feature is a 'tone' and each row forms a 'chord' of tones. The 10x10 grid generates 10 chords, one for each row, such as "foliage sky" or "grass sky sand water"[2] etc.

4) A binary 'Touching' relationship (additional to the above 3 representations already used in our work [13]), which records whether one patch type is spatially in contact with another in the image. Note that, although apparently similar, the Allen 'meets' relation is not equivalent since the 2 patches may be at different sides of the picture.

For comparison purposes, we also ran experiments with a purely quantitative metric based retrieval scheme based on the respective percentages of each of the semantic concepts in each image in the style of [19]. This representation is labelled as "Percentages" in Table 1. Similarity is computed using the sum of absolute differences in percentage values for each attribute in a pair of images.

Fig. 1(b) illustrates the chord representation while Fig. 1(a) the relative size and Allen relationships. Several variants of the above QSRs were also investigated; we report

[1] There are 9 semantic concepts in [19], while the data set provided and which has been used in our experiments contains 2 extra ones (mountain and snow) – however these occur infrequently and the basis for comparison will be thus essentially unaffected.

[2] This representation can be regarded as an abstraction of the relation used by [19] – whereas they record the percentage of each attribute in each horizontal band, in the chord representation it is only the presence or absence which is recorded.

on just one here where the relative size representation is recorded separately within 3 image areas: Top (T: top 3 rows), Middle (M: rows 4-7), Bottom (B: rows 8-10).

(a) (b)

Fig. 1. QSR using (a) relative size and Allen's calculus (b) chord representation

4. IR Based on Qualitative Similarity

We envisage a CBIR system in which a query is specified either by giving an example image or by a symbolic query expressed in terms of the qualitative relations defined above, e.g. "retrieve images with rocks touching water and more water than foliage". In the former case, we can compute a qualitative description of the image using one more of our qualitative schemes, but in this case it is more likely that no image will exactly match – this could also happen in the latter case. It would clearly be convenient to be able to retrieve images which nearly match the query (which ever way it is specified). The problem is to define what "nearly matches" means, since in a qualitative representation we do not have raw numbers available. In the remainder of this section we define notions of qualitative similarity for each the qualitative representations.

The CN of Allen relations is presented in Fig 2(a) below. The links connect neighbouring relations – ones which are most similar – as one traverses more links from a particular relation, the relations become progressively less similar. Thus if in image 1 sky **<** grass, and also in image 2, then they are identical (in this comparison); if in image 3 sky **m** grass, then image 3 is similar to image 1, whilst if image 4 has sky **o** grass, then image 4 is also similar to image 1 but not as similar as image 3, and so forth. Since there are many attributes in each description of an image (e.g. 66 in Allen representation), we have to find a way to combine the similarities of each pairwise comparison. The CN for the Allen relations is already a partial order, and it is clear that the cross product across all the attributes is even more so. To achieve a total ordering we assign a weight of 1 to each arc in the CN, and sum the number of arcs traversed across all the attributes in order to transform one description into another (using the shortest route). Clearly we could assign non uniform weights to the different arcs but in the absence of any particular reason to do this, a uniform weighting appears to be the obvious choice. The situation where one of the relations from a particular pair of images for a pair of attributes is "no" whilst the other is not, deserves some discussion – what should be the weight in this case (since "no" does not appear in the CN)? One possibility is to choose a weight of 7 (one more than the maximum weight otherwise in the Allen CN), though other choices could clearly also be used, and indeed we also experimented with the choice of zero[3] and values greater than and less than 7. In an implementation for an end user, this could be a parameter (perhaps a slider in the interface).

[3] This was particularly motivated by classes such as "lwm" where the set of concepts present can vary considerably, and penalizing image with a different set of concepts to the query image had a great effect on the results. A penalty weight of 0 implies that the similarity of images is determined only by the relationship between common semantic concepts in the query and database images, and missing concepts do not contribute towards total penalty weight.

The CN for the relative size representation is much simpler with just three nodes, one for each of the three relations, with ≈ neighbouring each of $<$ and $>$ and the maximum weight is 2. For missing patch types we do not need a 'no' relation in this representation since their size is zero and the existing three relationships are still applicable.

For the case of the chord representation, we can think of the CN as being equivalent to a complete lattice generated by the power set of the set of patch types; effectively this means that the similarity is directly proportional to the number of insertions and deletions required to transform one chord into another.

For the representation of spatial touching, there are just 2 nodes in the CND (touching and not-touching) and a single link connecting them. We experimented with this representation, however eventually used a similarity measure which also takes account of the degree of touching. Each patch in the rectangular grid can touch up to 8 other patches. For a pair of given patch types p1 and p2, we compute how many patches of type p1 touch a patch of type p2, and vice-versa for p2 and p1; the maximum of these 2 values is then recorded as one of the attributes in this representation of an image. To compute the degree of similarity between two images using this representation we simply take the sum of the absolute differences in each of the corresponding attribute values for each image. This representation thus combines a very qualitative representation, touching, which is a purely topological relationship, with a metric measurement of its applicability to a particular image. Thus, for example, for an image with extended sky-grass spatial connection will be more similar than ones with small amount of spatial connection between the two concepts.

Thus given a representation "R" with attributes $\mathrm{A}_1^{\mathrm{R}} \ldots\ldots \mathrm{A}_{|\mathrm{R}|}^{\mathrm{R}}$, and a function $f^R(u,v)$ which gives the similarity between two attribute values u and v then the overall similarity S^R (x,y) between two images *x* and *y* in representation 'R' is given by:

$$S^R(x,y) = \sum_{i=1}^{i=|R|} f^R(A_i^R(x), A_i^R(y)) \qquad (1)$$

We then can compute rank of an image *y* in the database for query image *x* as:

$$Rank^R(x,y) = |\{z : S^R(x,z) < S^R(x,y)\}| \qquad (2)$$

5. Results and Evaluation

We have conducted experiments with each of the representations above individually and also in various combinations. To illustrate the results obtained, we first present (fig. 2(b)) a sample query image and the top 5 results according to the qualitative similarity measures described in §4 for Allen representation. This does not give any quantitative evaluation of the quality of the retrieval and we next turn to this question. To provide a more thorough quantitative analysis of the performance of the various representations, we used the following experimental setup. Each of the 700 images in the database was used as a query image in turn, and a similarity ordering computed for all the other 699 images. However this does not tell us whether images high in the ordering really are intuitively similar to the query image. As a proxy for an extensive user evaluation of each of these rank orderings, we use the hand assigned category labels used for previous work on this dataset for supervised learning of category descriptions [19,13].

Given a query image in category *c*, we can evaluate the number and hence the percentage of images in the same category in the top *k* images in the rank ordering. For cases where the number of images of a particular category in the DB is less than *k* clearly 100% scores cannot be achieved.

The number *k* may be user defined, or be determined by conditions such as how many images of a certain size fit on a user's screen, or could be determined by analysis of the actual similarity values. Table 1 shows, for each class, the number of retrieved images of that class in the top ranked 20 and the top *k* images (where *k* is the number of images in the respective class, e.g. *k*=34 for sky_clouds), each row giving the values for a different representation. The last two rows in Table 1 shows the statistics when using the percentage of each semantic attribute as the representation for comparison with the quantitative techniques of [19]. The results reveal the following interesting conclusions:

- The recall rate clearly validates the measures of similarity used, since as the number of images retrieved increases, the accuracy of retrieved images goes down (measured by successive retrieved images of the same category).
- the recall percentages are well above the baseline statistical likelihood of each category of images in the population.
- The chord representation performs relatively well. Arguably this is because it closely resembles the human cognition of similarity because a human may describe or compare an image in terms such as "having sky in the top, foliage and water in the middle, water and sand at the bottom of image" – remembering that the semantic categories were assigned by a human (though without being aware of the possibility of subsequently using the chord representation (or indeed any other).

(a) (b)

Fig. 2. (a) CN for Interval Calculus [42] **(b)** Query & top 5 retrievals using Allen's Rep

- The representation 'relative size' performs surprisingly well, given the low information content. Moreover, the relative size on TMB regions of image representation performs at least as well if not even better in overall compared to the purely metric representation (Percentages and Percentages on TMB).
- The touch based representation does not perform particularly well – arguably it does not encode sufficient information to be able to adequately distinguish cognitive similarity in the image dataset.

Table 1 only considers individual representations. Since the performance of representations varies across categories (and bearing in mind that we assume we do not know the category of an image – we are using this information here purely for evaluation purposes), we also experimented with similarity measures based on combinations of four different qualitative representations[4] – Allen, relative size, chord and touching.

There have been a number of approaches in image categorization research involving bagging/boosting while in IR, multiple query processing or use of low level and semantic labels has been used to improve the retrieval accuracy. We investigated voting approaches based on combining the respective penalty weights of images in individual representations, and on combining the ranks of retrieved images in each selected QSR.

[4] Of course each representation might itself be viewed as a hybrid representation with the 66 attributes (or whatever number of attributes used in the particular representation) combining together to assign an overall similarity to an image pair.

In order to count the accumulative effect of penalty weights in all of the 4 selected representations and also the overall ranking of an image in the list of database images, several other kinds of weighted voting schemes ($V_1 - V_4$) were investigated (Table 2):

V_1- Compute:

$$S^{V_1}(x,y) = \sum_{r=1}^{r=4} S^r(x,y) \quad (3)$$

for each image in the DB for a query *x* and then sort in ascending order.

V_2- Compute:

$$S^{V_2}(x,y) = \underset{r=1}{\overset{r=4}{Min}}\ S^r(x,y) \quad (4)$$

for each image in the DB for a query *x* and then sort in ascending order: (variant of V1).

Although the weights within in each representation may be regarded as comparable, it is arguable as to whether this also holds with respect to the weights in other representations. We thus investigated schemes based solely on the rank within each of the four representations.

V_3- Compute:

$$S^{V_3}(x,y) = \sum_{r=1}^{r=4} rank^r(x,y) \quad (5)$$

for each image in the DB for a query *x* and then sort in ascending order.

V_4- Compute:

$$S^{V_4}(x,y) = \underset{r=1}{\overset{r=4}{Max}}(rank^r(x,y)) + \underset{r=1}{\overset{r=4}{Max2}}(rank^r(x,y)) \quad (6)$$

where "Max" and "Max2" compute the maximum and 2nd highest values respectively.

The results suggest the following conclusions:

- The purely qualitative approaches perform comparably or even slightly better in some cases to the quantitative ones. The former have added advantage that they also allow retrieval based on simple linguistic descriptions using qualitative descriptions over the semantic attributes.
- The voting schemes based on accumulative weighted votes and weighted rank votes ($V_1 - V_4$) perform better than the approaches using a single representation only.
- The overall accuracy of the retrieval process compared with the actual class labels is somewhat problematic due to the fact that many images may be categorized as either "lwm" or "coast" – i.e. most of the images in the DB have some aspects of "lwm" or "coast", and arguably it is a matter of degree or personal preference when an lwm with sky above becomes a "sky_clouds". Similarly, there is lot of potential confusion in images categorised in classes like "fields" and "sky_clouds". This fact was also established in [13, 19] while learning the class descriptions.
- The voting schemeV_1 performs much better in the top 20 and the top *k* experiments as it is based on accumulative row weights of an image corresponding to 4 representations chosen. Its performance is comparable to the quantitative approach. Furthermore, both of the basic voting schemes, V_1 and V_3, are better than the individual representations in terms of accuracy of IR using the "ground truth" of the hand assigned labels.
- It can be seen that coasts and waterscapes do relatively badly compared to the other categories, and this is also true about sky_clouds and fields categories in some of the representations, which is not altogether surprising from a semantic/intuitive viewpoint. If these two categories are combined into a single category then the rate of accuracy improves significantly. This fact has also been observed in the confusion matrices of different learning schemes in [13].

Categories / QSRs	Coasts Out of		Field Out of		Forest Out of		LWM Out of		Sky_Clouds Out of		wscapes Out of		Overall	
	20	**k**	**20**	**k**	**20**	**k**	**20**	**k**	**20**	**k**	**20**	**k**	**20**	**k**
Allen only	56	33	38	26	66	41	84	48	49	35	46	26	59	36
Touch	57	33	40	27	73	51	85	52	51	40	42	22	61	38
Chord	56	41	66	34	91	68	82	59	91	**89**	47	36	70	50
Size only	63	**46**	57	34	86	66	**88**	61	60	44	**51**	**37**	70	49
Size on TMB	**67**	45	**68**	**38**	**92**	**75**	88	**65**	**93**	82	47	34	**74**	**53**
Percentages-%s	62	47	**70**	36	92	69	**84**	61	93	91	47	**36**	**73**	52
%s on TMB	**64**	**48**	69	**36**	**93**	**72**	84	**62**	**94**	**92**	**48**	35	73	**53**

Table 1. Recall percentages on per category and overall basis in top 20 & number of images in each category (k) for all representations used.[5]

Categories / QSRs	Coasts Out of		Field Out of		Forest Out of		LWM Out of		Sky_Clouds Out of		Wscapes Out of		Overall	
	20	**k**	**20**	**k**	**20**	**k**	**20**	**k**	**20**	**k**	**20**	**k**	**20**	**k**
V_1	**67**	**45**	**69**	**35**	**95**	**78**	92	**69**	**88**	**78**	**51**	**35**	**76**	**54**
V_2	55	33	37	26	65	42	83	48	50	35	47	27	59	36
V_3	66	44	60	33	93	72	**93**	65	79	63	50	33	**74**	**51**
V_4	66	42	60	34	87	64	90	60	69	48	**51**	33	72	47

Table 2. Recall percentages on per category and overall basis in top 20 & number of images in each category (k) for weighted voting schemes.

6. Conclusions And Further Work

We have presented an approach to CBIR based on semantic knowledge and QSR. The approach does not rely either on segmentation techniques applied directly or on low level image features for an image description. We have presented similarity measures of the qualitative spaces based on the conceptual neighbourhoods that typically accompany qualitative calculi and experimental results for IR using a variety of qualitative description languages and several combinations of these. We are not necessarily arguing that these are the best languages either for this particular data set or in general. It is the overall approach we present which we believe is the most important result of this research, which shows that qualitative representations can rival metric ones, whilst providing more intuitive descriptions. We have also presented a variety of voting schemes for combining representations and evaluated their success on the image dataset. The evaluation was based on a hand labelled categorization which although it has some disadvantages, does provide a cognitive basis for evaluating the retrieval results. It may be noted that in all cases, the recall percentages are well above the baseline statistical likelihood of each category of images in the population.

A variety of further work suggests itself including the evaluation on other data sets, using actual user analysis to evaluate the results (cf the psychophysical experiments in [19]), experimentation with other qualitative calculi, and combining qualitative and quantitative representations. We already have a prototype user interface to an IR system based on the ideas presented here; this could be further improved to provide a flexible interface based on query by image or by qualitative description, or a combination of the two, with the user free to select the kinds of descriptions, similarity measures and voting

[5] Bold figures in Table 1 and Table 2 indicate best ones in qualitative and quantitative representations, while k=143,128,103,178,34 and 114 for above mentioned six classes – in order as these appear in table.

schemes most appropriate to their needs. The analysis here provides the basis for reasonable default choices.

Acknowledgements: We thank Julia Vogel for providing the labelled data set and helpful discussions and acknowledge financial support provided by National University of Sciences & Technology, Rawalpindi – Pakistan, and EPSRC grant EP/DO61334/1 to Zia Ul Qayyum and A.G. Cohn, respectively.

References

[1] Aghrabi, Z. , Makinouchi, A. "Semantic Approach to Image Database Classification & Retrieval". NII journal,. 7 (.9), 2003.
[2] Allen, J. F. "Maintaining knowledge about temporal intervals". C of ACM, 26(11), 1983.
[3] Bonan, Li, and Fonsesca, F. "TDD: A Comprehensive Model for Qualitative Spatial Similarity Assessment". In J. of Spatial Cognition and Computation, 6(1), pp.31-62, 2006.
[4] Bradshaw, B. "Semantic Based Image Retrieval: A Probabilistic Approach." ACM Multimedia, October 2000.
[5] Burns, H.T., Egenhofer, M.J. "Similarity of Spatial Scenes". J.-M. Kraak and M.Molenaar (Eds), 7th Int Symp on Spatial data Handling, Taylor & Francis, London, pp. 173-184, 1996.
[6] Cohn, A G and Hazarika, S M. "Qualitative Spatial Representation and Reasoning: An Overview". Fundamenta Informaticae, 46(1-2), pp. 1--29, 2001.
[7] Deb, S., & Zhang, Y. "An overview of Content-based Image Retrieval Techniques". Proc 18th Int. Conf on Advanced Information Networking & Application , 2004, pp. 59-64.
[8] Depalov, D, Pappas, T, Li, D, & Gandhi, B, "Perceptually Based Techniques for Semantic Image Classification & Retrieval". Human Vision & Electronic Imaging, XI (B.E. Rogowitz, T.N. Pappas, & S.J. Daly Eds.), Proc. SPIE,. 6057, CA, 2006.
[9] Enser, P., Sandom, C., "Towards a Comprehensive Survey of the Semantic Gap in Visual Image Retrieval." Springer LNCS, Vol. 27-28, pp. 279-287, 2003.
[10] Freksa, C. "Temporal Reasoning Based on Semi-intervals." Art. Int., 54(1-2), 199-227.
[11] Howe, N. "A Closer Look at Boosted Image Retrieval". Int. Conf on Image & Video Retrieval, Vol. 2728, LNCS, pp. 61-70, 2003.
[12] Hyvonen, E., Styrman, A., and Saarela, S. "Ontology-Based Image Retrieval". HIIT publications, No. 2002-03, pp. 15-27, Helsinki Institute of IT, Helsinki, Finland, 2002.
[13] Qayyum, Z.U. and Cohn, A.G. "Qualitative Approaches to Semantic Scene Modelling and Retrieval". In Proc. Of SGAI-AI'06, Research and Development in Intelligent Systems XXIII, Springer-Verlag, 2006.
[14] Rasiwasia, N., Vasconcelos, N. and Moreno, P.J. "Query by Semantic Example". H. Sundaram et al (Eds.): CIVR 2006, LNCS 4071, pp. 51-60, 2006.
[15] Sebe, N., Lew, M.S., Zhou, X., Huang, T.S., and Bakker, E.M. "The State of the Art in Image and Video Retrieval". Springer LNCS, Vol. 2728, pp. 1-8, 2003.
[16] Sun, J.Y., Sun, Z.X., Zhou, R.H., and Wang, H.F. "A Semantic-Based Image Retrieval System: VISENGINE". Proc. 1st Int. Conf on Machine Learning and Cybernetics, 2002.
[17] Town, C., and Sinclair, D. "Content-based image retrieval using semantic visual categories". Tech. Report 2000.14, AT&T Laboratories Cambridge, 2000.
[18] Veltkamp, R.C., and Tanase, M. "Content-Based Image Retrieval Systems: A Survey". Univ. Utrecht, Utrecht, The Netherlands, Tech. Rep. UU-CS-2000-34.
[19] Vogel, J., and Schiele, B. "Semantic Modelling of Natural Scenes for Content-Based Image Retrieval". Int J of CV, Springer , 10.1007/s 11263-006-8614-1, 2006.
[20] Wang, W., ,Song,Y., Zhang, A. "Semantics-Based Image Retrieval By Region Saliency". LNCS, Vol. 2383, Proc. Int. Conf on Image and Video Retrieval, pp. 29 – 37, 2002.
[21] Wang, W., Song, Y., Zhang, A. "Semantic Retrieval by Content and Context of Image Regions". Proc 15th Int. Conf on Vision Interface (VI'02), 2002.

Conditional Random Field for Natural Scene Categorization

Yong Wang and Shaogang Gong
Department of Computer Science
Queen Mary, University of London
{ywang,sgg}@dcs.qmul.ac.uk

Abstract

Conditional random field (CRF) has been widely used for sequence labeling and segmentation. However, CRF does not offer a straightforward approach to classify whole sequences. On the other hand, hidden conditional random field (HCRF) has been proposed for whole sequences classification by viewing the segment labels as hidden variables. But the objective function of HCRF is non-convex because of its hidden variable structure. In this paper, we propose a classification oriented CRF (COCRF) adapted from HCRF for natural scene categorization by taking an image as an ordered set of local patches. Our approach firstly assigns a topic label to each segment on the training data by the probabilistic latent semantic analysis (PLSA) and train a COCRF model given these topic labels. PLSA provides a higher level of semantic grouping of image patches by considering their co-occurrence relationships while COCRF provides a probabilistic model for the spatial layout structure of image patches. The combination of PLSA and COCRF can not only classify but also interpret scene categories. We tested our approach on two well-known datasets and demonstrated its advantage over existing approaches.

1 Introduction

This paper addresses the problem of natural scene categorization. Scene understanding underlies many other problems in visual perception such as object recognition and environment navigation. Although scene categorization can be achieved at a glance by a human, it poses great challenges to a computer vision system. Different instances of the same category can vary a lot in their color distribution, texture patterns and more importantly, a scene category does not have a well-defined shape as an object category does.

Recent work in scene image classification focus on image classification based on an intermediate level of features. They can be further divided into two categories. The first relies on self-defining the intermediate features. Oliva and Torralba [7] proposed a set of perceptual dimensions (naturalness, openness, roughness, expansion and ruggedness) that represent the dominant spatial structure of a scene. Each of these dimensions can be automatically extracted and scene images can then be classified in this low-dimensional representation. Vogel and Schiele [8] used the occurring frequency of different concepts (water, rock, *etc*) in an image as the intermediate features for scene image classification,

and they need manual labeling of each image patch in the training data. While manual labeling can improve the semantic interpretation of images, it is still a luxury for a large dataset and it can also be inconsistent in defining a common set of concepts [8]. The second kind of approach is aimed to alleviate this burden of manual labeling and learn the intermediate features automatically. This is achieved by making an analogy between a document and an image and taking advantage of the existing document analysis approaches. For example, Fei-Fei and Perona [2] proposed a Bayesian hierarchical model extended from latent dirichlet allocation (LDA) to learn natural scene categories. Bosch et al. [1] achieved good performance in scene classification by combining probabilistic latent semantic analysis (PLSA) [3] and a KNN classifier. A common point of these approaches is that they represent an image as a bag of orderless visual words. An exception is the work done by Lazebnik et al. [6] where they proposed spatial pyramid matching for scene image classification by partitioning an image into increasingly fine sub-regions and taking each sub-region as a bag of visual words.

As a simple but discriminative enough representation, the bag of visual words has shown its advantage in the above approaches. However, its assumption of an orderless bag makes it inevitably sacrifice certain amount of discriminative capability. The order statistics are actually quite helpful in our understanding of scenes. At least two cues can be applied. The first is the spatial layout of the patches. For example, *sky* always appear in the upper part of an image and *ground* almost always appear in the bottom part. Lazebnik et al. [6] have demonstrated the advantage of this cues, but they did not do it in a probabilistic model. The second cue is the spatial pairwise interaction between two neighboring patches. For example, it is more likely to find a *water* patch as the neighbor as a *sand* patch in a *beach* scene, while in a *coast* scene water patches are usually adjacent to *stone* patches. None of the existing approaches have modeled both of these two relations explicitly in a probabilistic model.

A good candidate for modeling a set of ordered local patches is the conditional random field (CRF) [5]. For example, Kumar and Hebert [4] attempted to use a discriminant random field to model contextual interaction between image patches. But their work was for image region classification, instead of whole image classification. Generally speaking, CRF is aimed for segment labeling and segmentation. It does not offer a straightforward approach to classify whole sequences and requires the labeling of the segments in the training data. Hidden conditional random field (HCRF) [9] was proposed for whole sequences classification by viewing the segment labels as hidden variables, but the hidden variable structure makes the objective function of HCRF non-convex and only local optimum can be achieved in training. In this paper, we proposed a combinational approach of PLSA and a classification oriented CRF (COCRF) adapted from HCRF for natural scene categorization by taking an image as an ordered set of image patches. COCRF takes the advantage of automatic labels generated by PLSA and is capable of reaching a global optimum in the training stage. The motivations of PLSA here are not only that it can provide labeling of the image patches, but also that it is complimentary to COCRF, i.e., PLSA can discover the *co-occurrence* relationship between image patches, while COCRF can only model *spatial* relation between patches. Thus our PLSA+COCRF model can take into account both of these two factors. An obvious advantage of our approach is to provide a probabilistic way to model both the spatial layout of image patches and their neighboring interaction. We tested our approach on two scene image image datasets and show that it outperforms existing approaches.

The rest of this paper is organized as follows. Section 2 describes the topic labeling of image patch by PLSA. Section 3 introduces COCRF and focus on the features we have deployed. Section 4 discusses the learning and inference of COCRF for classification. We show some experimental results in section 5 and conclude in section 6.

2 Automatic Topic Labeling of Image Patches via PLSA

In our approach, an image is represented as a number of image patches. Each patch is assigned a topic label automatically through PLSA [3]. PLSA can be summarized as follows. Suppose we have a collection of text documents $\mathscr{D}=\{d\}$, a vocabulary $\mathscr{W}=\{w\}$ and a number of topics $\mathscr{S}=\{s\}$. Each document d is represented as a bag of words, i.e, we keep only the counts $n(d,w)$ which indicates the number of occurrence of word w in document d. PLSA assumes that each word in a document is generated by a specific topic. Given the topic distribution of a document, its word distribution is independent from the document. More precisely, the probability of a word w in a document d is a marginalization over topics, i.e.,

$$P(w|d) = \sum_{s\in\mathscr{S}} P(w|s)P(s|d) \tag{1}$$

Given $\mathscr{D}$ and $P(w|d)$, the parameters $P(s|d)$ and $P(w|s)$ can be estimated by an EM algorithm [3]. To adapt PLSA to image data, we transform images into the bag of visual words representation by the following procedures: (i) Partition each image into a number of small patches. (ii) Learn a visual vocabulary on the descriptors of a subset of local patches by k-means clustering. (iii) Assign a visual word to each local patch. After a PLSA model is learned from the training images, we can obtain the topic labeling s of a visual word w in a specific document d by the following equation

$$P(s|w,d) = \frac{P(w|s)P(s|d)}{P(w|d)} \tag{2}$$

The ending results of PLSA is that each image patch has a topic label.

3 Classification Oriented Conditional Random Field (COCRF)

Our final objective is to assign a scene category label to a given image. The training data is $\{(y^{(k)},\mathbf{x}^{(k)},\mathbf{s}^{(k)})\}$, where $y^{(k)}$ is the category label, $\mathbf{x}^{(k)} = \{x_1^k, x_2^k, x_{n_k}^k\}$ are the visual features of each image patch, $\mathbf{s}^{(k)} = \{s_1^k, s_2^k, s_{n_k}^k\}$ are the corresponding topic labels of the image patches obtained by PLSA. k is the index of the training image. The graphical structures of CRF, HCRF and COCRF are illustrated in Fig. 1. In these graphic models, we have taken an image with four local patches (which we also refer to as segments) as an example. The scene category label is denoted by variable y and $\mathbf{s} = \{s_1, s_2, s_3, s_4\}$ are the topic labels of the image patches. The image observation is denoted by variables $\mathbf{x} = \{x_1, x_2, x_3, x_4\}$. The edges between nodes represent their inter-dependence. The shaded nodes in HCRF indicate these nodes are hidden variables. In our model, we consider the graphic structure of nodes $\mathbf{s}$ as a lattice with pairwise potentials. In a CRF model, we have only the topic labels and the image observation. In HCRF we have an additional node y but $\mathbf{s}$ is not observed. In COCRF we have the node y and all the nodes $\mathbf{s}$ are observed.

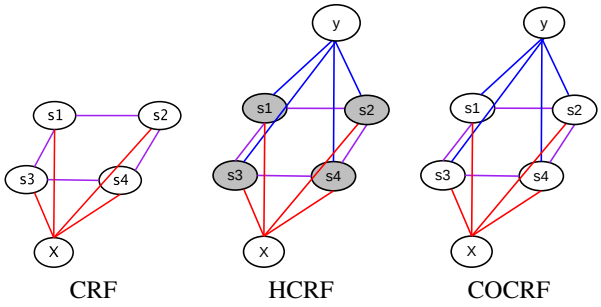


Figure 1: Graphical models of conditional random field (CRF), hidden conditional random field (HCRF) and classification oriented conditional random field (COCRF).

Following the definition of a CRF model, the conditional probability for the topic labels $\mathbf{s}$ and the category label y given the observation $\mathbf{x}$ can be expressed as

$$P(y,\mathbf{s}|\mathbf{x};\theta) = \frac{e^{\psi(y,\mathbf{s},\mathbf{x};\theta)}}{\sum_{y',\mathbf{s}'} e^{\psi(y',\mathbf{s}',\mathbf{x};\theta)}} \tag{3}$$

where θ represents the parameters of the model. $e^{\psi(y,\mathbf{s},\mathbf{x};\theta)}$ is the potential function. In COCRF, we consider three types of potential and we write the log potential function $\psi(y,\mathbf{s},\mathbf{x};\theta)$ as the summation of three terms. Each term can be viewed as a different type of features deployed for classification.

$$\psi(y,\mathbf{s},\mathbf{x};\theta) = \underbrace{\psi^a(y,\mathbf{s},\mathbf{x};\theta)}_{\text{node appearance potential}} + \underbrace{\psi^e(y,\mathbf{s},\mathbf{x};\theta)}_{\text{edge potential}} + \underbrace{\psi^s(y,\mathbf{s};\theta)}_{\text{node spatial potential}} \tag{4}$$

3.1 Appearance Potential

The appearance potential measures the compatibility between a topic label and its appearance. This potential is a kind of low-level features and it is shared among different scene categories.

$$\psi^a(y,\mathbf{s},\mathbf{x};\theta) = \sum_{j=1}^{m} \phi(\mathbf{x},j)\cdot\theta^a(s_j) \tag{5}$$

where j is the index of a segment (patch) and m is the total number of segments. $\phi(\mathbf{x},j) \in \mathbb{R}^d$ is a feature extraction function which maps the observation at site j to a d-dimensional feature vector. $\theta^a(s_j)$ is the appearance parameter vector corresponding to the segment label $s_j \in \mathscr{S}$.

Considering the diversity in appearance of each topic, we map the local observation to a feature vector by a Gaussian Mixture Model (GMM). Suppose we have a set of Gaussian components $\{g_1, g_2, \ldots, g_d\}$, each of which has its own parameters of the mean and variance. The feature extraction function is represented as,

$$\phi(\mathbf{x},j) = \left[g_1(x_j), g_2(x_j), \cdots, g_d(x_j)\right]^t \tag{6}$$

where x_j is the appearance descriptor of segment j. To obtain the set of Gaussian components $\{g_1, g_2, \ldots, g_d\}$, we firstly collect a subset of local patches of each topic and fit a GMM to each topic. The final set of Gaussian components are the combination of all the Gaussian components for each topic.

3.2 Edge Potential

The edge potential models the interaction between neighboring patches. It is similar to that in CRF but it is category dependent. This provides COCRF more discriminative capability between different categories, as follows

$$\psi^e(y,\mathbf{s},\mathbf{x};\theta) = \sum_{(j,k)\in E} \theta^e(s_j,s_k,y) \tag{7}$$

where θ^e is symmetric with respect to s_j and s_k. E is the set of all the edge links between the segment nodes depending on the 2-D lattice structure.

3.3 Spatial Layout Potential

Here we take an explicit approach by dividing the image area into 3× 3=9 sub-regions. We examine the the spatial layout distribution of each topic on this 3×3 grid.

$$\psi^s(y,\mathbf{s};\theta) = \sum_{j}^{m} \theta^s(y,s_j,\eta(j)) \tag{8}$$

where $\eta(j) \in \{1,2,\ldots,9\}$ denotes the deterministic mapping function of a site j into the sub-region it sits in. It is worth noting that if θ^s does not depend on the spatial location of node j, this potential will degrade to the one as same as that in HCRF [9].

4 Learning

In the training process we learn the model parameter $\hat{\theta}$ by maximizing its log likelihood on the training data. Assume the training data is *i.i.d.*, $\hat{\theta}$ is obtained by,

$$\hat{\theta} = \arg\max_{\theta} \mathscr{L}(\theta) = \arg\max_{\theta} \sum_{k=1}^{n} \mathscr{L}^k(\theta) \tag{9}$$

where $\mathscr{L}^k(\theta)$ is the log likelihood of the k-th sample and n is the total number of training samples. Since $\mathbf{s}^{(k)}$ is observed, we have

$$\mathscr{L}^k(\theta) = \log P(y^{(k)},\mathbf{s}^{(k)}|\mathbf{x}^{(k)};\theta) = \log\left(\frac{e^{\psi(y^{(k)},\mathbf{s}^{(k)},\mathbf{x}^{(k)};\theta)}}{\sum_{y',\mathbf{s}'} e^{\psi(y',\mathbf{s}',\mathbf{x}^{(k)};\theta)}}\right) = \psi(y^{(k)},\mathbf{s}^{(k)},\mathbf{x}^{(k)};\theta) - \log\sum_{y',\mathbf{s}'} e^{\psi(y',\mathbf{s}',\mathbf{x}^{(k)};\theta)} \tag{10}$$

This equation is different from that in HCRF [9], where the topic labels $\mathbf{s}^{(k)}$ have to be marginalized out because they are not observed. Unlike HCRF, $\mathscr{L}^k(\theta)$ is concave because the first term is a linear function of θ and the second term is a log-sum-exp which is convex. The optimization is based on the quasi-newton algorithm, so we need the first-order derivatives of the log likelihood with respect to the model parameters θ. For convenience, we reformulate $\psi(y,\mathbf{s},\mathbf{x};\theta)$ as a linear function of the model parameters [5, 9], i.e.,

$$\psi(y,\mathbf{s},\mathbf{x};\theta) = \sum_{j}\sum_{l\in L^1} \theta_l^1 f_l^1(j,y,s_j,\mathbf{x}) + \sum_{(j,k)\in E}\sum_{l\in L^2} \theta_l^2 f_l^2(j,k,y,s_j,s_k,\mathbf{x}) \tag{11}$$

where θ_l^1 is the clamped parameters of θ^a and θ^s. θ_l^2 is the clamped parameters [1] of θ^e. f_l^1 and f_l^2 are the corresponding binary feature functions. The dependency of f^1 and f^2 on site index j and k is for the general formulation. In our problem, we have only one feature function for nodes and edges respectively, i.e., $|L^1| = |L^2| = 1$. We consider the derivative with respect to the node potential parameters θ_l^1 based on this formulation. For simplicity, we omit the upper index k for a specific training sample so that $(y, \mathbf{s}, \mathbf{x})$ actually refers to $(y^{(k)}, \mathbf{s}^{(k)}, \mathbf{x}^{(k)})$. It can be derived that,

$$\frac{\partial \mathscr{L}^k(\theta)}{\partial \theta_l^1} = \sum_j f_l^1(j, y, s_j, \mathbf{x}) - \sum_{y', j, a} P(y', s_j{=}a|\mathbf{x}; \theta) f_l^1(j, y', a, \mathbf{x}) \tag{12}$$

Similarly, the derivative with respect to the edge potential parameters θ_l^2 can be written as

$$\frac{\partial \mathscr{L}^k(\theta)}{\partial \theta_l^2} = \sum_{(j,k)\in E} f_l^2(j, k, y, s_j, s_k, \mathbf{x}) - \sum_{y', j, k, a, b} P(y', s_j{=}a, s_k{=}b|\mathbf{x}; \theta) f_l^2(j, k, y', a, b, \mathbf{x}) \tag{13}$$

where,

$$P(s_j{=}a, y|\mathbf{x}; \theta){=}P(s_j{=}a|y, \mathbf{x}; \theta) P(y|\mathbf{x}; \theta) \tag{14}$$

$$P(s_j{=}a, s_k{=}b, y|\mathbf{x}; \theta){=}P(s_j{=}a, s_k{=}b|y, \mathbf{x}; \theta) P(y|\mathbf{x}; \theta) \tag{15}$$

By belief-propagation (BP) [10], we can calculate the two marginals in Eq. (14) and Eq. (15). As a by-product, BP can also calculate the partition function,

$$Z(y, \mathbf{x}; \theta) = \sum_{\mathbf{s}} e^{\Psi(y, \mathbf{s}, \mathbf{x}; \theta)} \tag{16}$$

so that we can calculate the marginal $P(y|\mathbf{x}; \theta)$ as

$$P(y|\mathbf{x}; \theta) = \frac{\sum_{\mathbf{s}} e^{\Psi(y, \mathbf{s}, \mathbf{x}; \theta)}}{\sum_{y', \mathbf{s}'} e^{\Psi(y', \mathbf{s}', \mathbf{x}; \theta)}} = \frac{Z(y, \mathbf{x}; \theta)}{\sum_{y'} Z(y', \mathbf{x}; \theta)} \tag{17}$$

Given the observation $\mathbf{x}$ of a new image and the learned parameter vector $\hat{\theta}$, we infer its category label $\hat{y}$ by maximizing the posterior probability. Since predicting the class label $\hat{y}$ is our ultimate goal, we marginalize out the topic labels $\mathbf{s}$, giving out

$$\hat{y} = \arg\max_y \sum_{\mathbf{s}} P(y, \mathbf{s}|\mathbf{x}; \hat{\theta}) = \arg\max_y P(y|\mathbf{x}; \hat{\theta}) \tag{18}$$

As noted in the above section, this can be efficiently calculated by BP.

5 Experiments

5.1 Datasets

We used two well known scene image datasets for our experiments: the Oliva and Torralba [7] dataset which we referred to as the OT dataset, and the Vogel and Schiele [8] dataset,

[1] The whole set of parameter is represented by a vector and the vector again is divided into blocks. The parameters in the same block can be updated together. *Clamped* means several parameters are put in the same block, this is for the convenience of implementation.

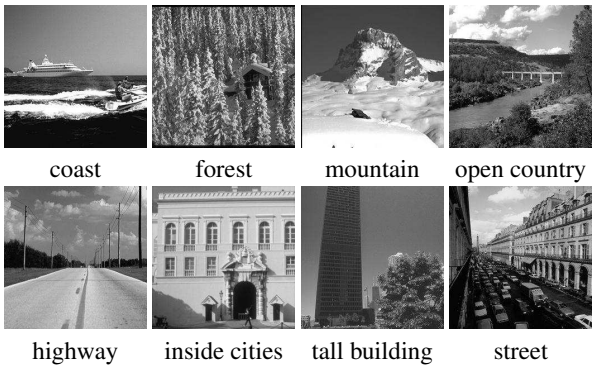


Figure 2: Sample images from the OT datasets.

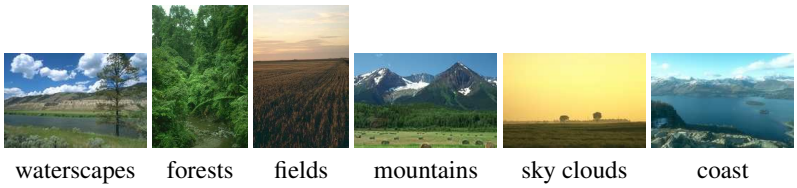


Figure 3: Sample images from the VS datasets.

referred to as the VS dataset. The OT dataset contains grayscale images of 8 scene categories. The category labels and the number of images of each category (in brackets) are: coasts (360), forest (328), mountain (374), open country (410), highway (260), inside of cities (308), tall buildings (356) and streets (292). All the images are in the same size as 250×250 pixels. The VS dataset contains 700 color images of 6 categories. The category labels and the number of images (in brackets) are: coast (142), waterscape (111), forest (103), field (131), mountain (179) and sky clouds (34). All the images in the VS dataset have been resized to 250 pixel in the maximum dimension. In Fig. 2 and Fig. 3 we show some sample images from these two datasets. Grayscale images are from the OT dataset and color images are from the VS dataset. We are aware that there are other datasets with more categories. The most complete set to our best knowledge is the 15 scene categories proposed by Lazebnik et al. [6], of which the OT dataset is only a subset. We have not chosen this one mainly because at this stage we have paid no effort on the speed of our algorithm. Working on the OT subset, we can have a more comprehensive evaluation. It is worth noting that although COCRF is computational more expensive compared to other approaches, it provides a probabilistic model to interpret the scene categories which other approaches cannot. The Bayesian approach by Fei-Fei and Perona [2] has this capability but they can not interpret the spatial layout structures of scenes.

5.2 Implementation

In our implementation, we partition each image into patches of 18×18 pixels and overlapping by 9 pixels. The number of patches of each image varies from 700 to 961. For

Table 1: Classification results in percentage on the OT and VS datasets.

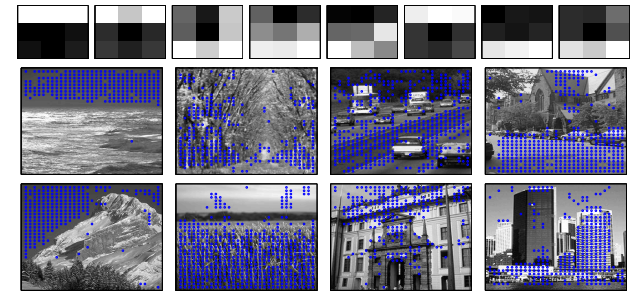
Performance on OT dataset					
Method	[1]	[6]	Task 1	Task 2	Task 3
Accuracy	86.65	86.85	82.3	87.13	90.2
Performance on VS dataset					
Method	[1]	[8]	Task 1	Task 2	Task 3
Accuracy	85.7	74.1	84.2	87.1	88.0

the grayscale images from OT dataset, we use SIFT descriptor as the feature vector for each patch. For the color images from VS dataset, we concatenated SIFT descriptor with another 6 dimensional color descriptor. The color descriptor represents the mean and variance of R,G and B. The visual vocabulary is generated by clustering a subset of 50000 image patches into 500 visual words on these two dataset respectively. PLSA is applied to group these visual words into 8 topics for both OT and OS. In generating the Gaussian components, the appearance of each topic is modeled by a mixture of 2 Gaussian components. Thus the final local appearance feature vector is a $2\times8=16$ dimensional vector. On the OT dataset, we take 100 images from each category for the training and the rest images for test (the same setup as [2] and [6]). On the VS dataset, we take half of the images from each category as training and the rest as testing (the setup as [1]). We have done several experiments including: (1) In task 1, we train COCRF with node potential but ignore the spatial location of each patch and edge potential. (2) In task 2, we train COCRF with spatial layout potential but without edge potential. (3) In task 3, we train COCRF with spatial layout potential and edge potential.

5.3 Results

Table 1 shows the classification results on the two datasets. The classification accuracy is calculated as the average of the classification accuracy of each category. In the following discussion we focus on the OT dataset. Task 1 is equivalent to take the number of occurrence of each topic in an image as the features and train a logistic classifier for image classification. Compared to the result (86.65%) in [1], our result (82.3%) in task 1 is a little worse. This is because their approach takes more training samples and trains a KNN as a non-linear classifier although the features are similar while ours is equivalent to a linear classifier. In task 2 we consider the number of occurrence of each topic and also the spatial layout of topics. This incorporation of spatial information of patches raise the recognition rate to 87.13%. It is better than that of [1] and [6] (86.65%). In [6], they also takes into account the spatial layout of each patches. Nevertheless, the result of their approach listed in Table 1 is conservative because we have taken out the classification accuracy of 8 categories from their 15 scene categories classification results. With less categories, the classification performance is expected to be slightly better. The best performance of of 90.2% is obtained in task 3. With 5 runs of task 3, each having a differnt partition of training and testing set, the deviation is 0.4%. This shows that the combination of spatial layout of individual patch and the pairwise interaction between patches is helpful for classification. The experimental results on the VS dataset shows the similar behavior.

As mentioned before, a benefit of COCRF is that it can discover the spatial layout



coast	forest	highway	street
mountain	open country	inside cities	tall building
topic=6	topic=3	topic=8	topic=5

Figure 4: Spatial distribution of topics per category. Each column illustrates two scene categories and the spatial distribution of a specific topic. The blue dots superimposed on the images illustrated the location of those image patches labeled as the corresponding topics. See text for explanation. (This figure is best viewed in color).

distribution of local patches and their pairwise interaction for a category. The ability of probabilistic modeling can not be achieved by those approaches such as those of Bosch et al. [1] and Lazebnik et al. [6]. In Fig. 4 we illustrate the learned 3×3 spatial layout distribution of different topics in some categories. In this figure, we compare the spatial layout distributions of a specific topic of two categories in each column. The first row shows the two distribution probability maps of a certain topic for the two categories. For example, in the first row and the first column, we show the spatial layout distribution of topic 6 for a coast scene in the left and that for a mountain scene in the right. The second and third rows in each column show an instantiation for each category respectively. The blue dots superimposed on the images illustrated the location of those image patches labeled as the corresponding topics. The fourth row is the text description explaining which categories and which topic are compared. It is interesting to discover that topic 6 in the moutain scene has a special distribution (mass in left top and right top part of an image) while the same topic in a coast scene is more evenly distributed in the top part of an image. In Fig. 5, we show the pairwise interaction potential map between different topics for four categories. The intensity of the cell in row i and column j represents the probability of that topic i and topic j appear as neighbors to each other. Since in scene images, it is very common that the same topic appears as neighbors, we have depressed the pairwise interaction between two same topics (diagonal cells). This is to highlight the pairwise interaction potential between different topics. From this figure we can find that different categories can have very different pattern of pairwise interaction potential between patches.

6 Conclusion

We have presented a classification oriented conditional random field (COCRF) for natural scene categorization. COCRF is adapted from HCRF and is a fully observed model for classifying a whole sequence instead of labeling each segment of a sequence. Our

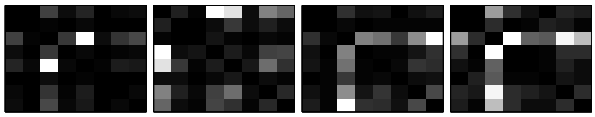


Figure 5: Illustration of the pairwise interaction potential between topics for four categories. The intensity of the cell in row i and column j represents the probability of that topic i and topic j appear as neighbors to each other.

approach is based on representing each image as an ordered set of local image patches. The training of COCRF needs both the topic labels and category labels of the training data. However, we do not need manual labeling of each segment. This is achieved by an automatic segment labeling process based on PLSA. PLSA can provide a higher level of semantic grouping of local patches by taking into account the co-occurrence relationship between different patches. COCRF provides a discriminative probabilistic model of the spatial layout of patches and their spatial pairwise interaction. Unlike HCRF, the objective function of training a COCRF model is convex, so we can avoid the concerns about local optimum and careful initialization. We have done experiments on two well-known scene image datasets. Our results demonstrate that COCRF outperforms the existing approaches for scene categorization.

References

[1] A. Bosch, A. Zisserman, and X. Munoz. Scene Classification via pLSA. In *Proceedings of the European Conference on Computer Vision*, 2006.

[2] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[3] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. In *Proc. of Uncertainty in Artificial Intelligence*, 1999.

[4] S. Kumar and M. Hebert. A discriminative framework for contextual interaction in classification. In *Proceedings of International Conference on Computer Vision*, 2003.

[5] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: probabilistic models for segmenting and labelling sequence data. In *Proceedings of International Conference on Machine Learning*, 2001.

[6] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[7] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, May 2001.

[8] J. Vogel and B. Schiele. Semantic modeling of natural scenes for content-based image retrieval. *International Journal of Computer Vision*, 72(2):133–157, 2007.

[9] S. B. Wang, A. Quattoni, L.-P. Morency, and D. Demirdjian. Hidden conditional random fields for gesture recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[10] Y. Weiss and W. Freeman. On the optimality of solutions of themax-product belief propagation algorithm in arbitrary graphs. *IEEE Transactions on Information Theory*, 47(2):723–735, 2001.

Evolutionary feature selection for probabilistic object recognition, novel object detection and object saliency estimation using GMMs.

Leonardo Trujillo [a], Gustavo Olague [a],
Francisco Fernández de Vega [b] and Evelyne Lutton [c]

[a] EvoVisión Project, Applied Physics Division, CICESE Research Center,
Km. 107 carretera Tijuana-Ensenada 22860, Ensenada, B.C. México.

[b] Grupo de Evolución Artificial, Universidad de Extremadura, Centro Universitario
de Mérida C/Sta Teresa de Jornet, 38, 06800 Merida, Spain.

[c] Complex Team, INRIA Roquencourt,
Domaine de Voluceau, BP 105, 78153 Le Chesnay Cedex, France.

{trujillo,olague}@cicese.mx,fcofdez@unex.es,evelyne.lutton@inria.fr

Abstract

This paper presents a method for object recognition, novel object detection, and estimation of the most salient object within a set. Objects are sampled using a scale invariant region detector, and each region is characterized by the subset of texture and color descriptors selected by a Genetic Algorithm (GA). Using multiple views of an object, and multiple regions per view, objects are modeled using mixtures of Gaussians, where each object represents a possible class for a particular image region. Given a set of objects, the GA learns a corresponding Gaussian Mixture Models (GMM) for each object in the set employing a *one vs. all* training scheme. Thence, given an input image where interest regions are detected, if a large majority of the regions are classified as regions of object *O* then it is assumed that said object appears in the imaged scene. The GA's fitness function promotes: 1) a high classification accuracy, 2) the selection of a minimal subset of descriptors, and 3) a high separation among models. The separation between two GMMs is computed using a weighted version of Fisher's linear discriminant, which is also used to estimate the most "salient" object among the set of modeled objects. Object recognition and *novel* object detection are done using confidence-based classification. Hence, when a non-modeled object is sampled, the detected regions are thereby identified as belonging to an unseen object and a new GMM is trained accordingly. Experimental results on the COIL-100 data set confirm the soundness of the approach.

1 Introduction

Currently, many computer vision systems address the problems of object detection and/or recognition using a sparse representation of image information through locally prominent

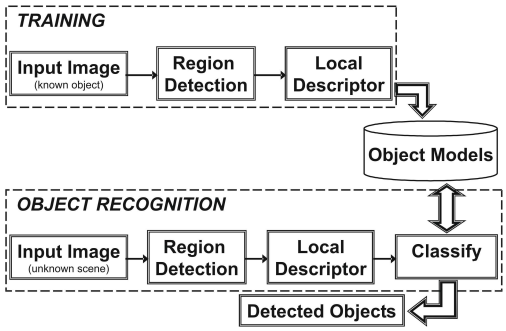


Figure 1: Abstract view of common object recognition vision systems.

image regions [8, 3], see Figure 1. A training phase consists on detecting stable image regions on an object using interest region detectors, and characterizing said regions using discriminative local descriptors [5, 3, 11, 10]. In this way, by relying on sparse local information the method is robust to partial object occlusions. During testing, an image is taken as input and the same region detection/description process is repeated. However, the extracted local information is now compared with stored object models and if appropriate matching criteria are met it is possible to identify known objects within the scene. This approach relies on the assumption that different local regions on an object will be highly separated in descriptor space, and thus requires highly discriminative region descriptors. This assumption will not hold true for objects with regular or repetitive patterns across their surface, i.e. a football or tomato. Furthermore, if object representations are learned in this manner, an intuitive comparison between two object models is not evident. For instance, if three object representations are learned, how can a measure of similarity be computed? These considerations are pertinent for a system that automatically identifies the "most salient" object, or image, from a given set. Automatic novelty detection is a line of research where these questions are essential [4]. Another application area relates to the automatic identification of visual landmarks; in robot navigation, for example, the norm is to use artificial or human selected landmarks.

This paper presents an approach where every region λ detected on an object O is taken as an instance of the same class, and is characterized with a feature vector of statistical descriptors computed in a feature space Φ of texture and color information. A GA searches within Φ for the smallest subspace $F \subseteq \Phi$ of statistical descriptors, of both texture and color, that yield the highest classification accuracy using a *one vs. all* scheme of maximum likelihood classification. The GA also searches for the best possible between-class separation of learned models. Therefore, the proposed approach does not require highly discriminative features because it uses a robust classifier, a known trade-off between descriptor design and classifier training. A GMM representation is used for each class (object), and a heuristic extension of Fisher's linear discriminant is used to estimate an "*apparent*" measure of class separation among models with more than one component. Based on this measure of model separation the most *salient* object is identified by selecting the object with the highest between-class separation using a min-max operation. A further advantage of using a GMM based classifier is the ability to use confidence estimation to identify regions extracted from unknown objects as outliers and label them as

samples of a new class. Hence, it is possible to automatically train a new *one vs. all* classifier for the newly identified object. Experimental results in this paper only deal with objects in scenes with simple backgrounds. Nevertheless, the use of a multimodal models should allow the approach to extend to real world scenes where more within class variation is likely to occur. Recently, Markou and Singh [4] propose a similar system that carries out both novelty detection and classification, however several differences exist:

1. The current work is concerned with object recognition, on the other hand, the work in [4] only addresses ROI classification.

2. The work in [4] relies on prior segmentation, a drawback because segmentation is an ill-posed problem; this is avoided by using locally salient image regions.

3. The proposed feature space Φ is more compact than the one used in [4], with less redundant information. Furthermore, the GA used for feature selection maximizes accurate classification, minimizes the set of descriptors used, and maximizes the between-class separation of learned models. The authors in [4] use the sequential floating forward selection algorithm and do not consider between-class separation.

4. The proposed measure for class separation is based on Fisher's linear discriminant which gives a closed form estimation computed directly from the learned GMMs; the Bhattacharya distance is employed in [4] along with NNet classifiers.

5. Novelty detection in the present work utilizes confidence-based classification of region descriptors, whereas [4] uses an heuristic criteria based on NNet output.

6. Finally, the COIL-100 data set used in the present work includes objects with information in feature space that tends to overlap, such as two toy cars with similar texture or two objects with the same color. On the other hand, [4] uses classes with marked differences among them, such as sky and chair classes.

2 Background

This section will give a brief review on some of the main concepts used throughout this work: scale invariant region detection, genetic algorithms, Gaussian mixture models, Fisher's linear discriminant, and the texture and color feature space employed.

Scale Invariant Region Detection. Selecting a characteristic scale for local image features is a process in which local extrema of a function response, embedded into a linear scale-space, are found over different scales. The interest operator applied in the current work was synthesized with Genetic Programming, optimized for high repeatability and global region separability [9, 10], named K_{IPGP1^*} which is based on DoG filtering,

$$K_{IPGP1^*}(\mathbf{x};t_j) = G_{t_j} * \left| G_{t_j} * I(\mathbf{x}) - I(\mathbf{x}) \right| , \tag{1}$$

where $j = 0,1,...,k$, and k is the number of scales to be analyzed, here it is set to $k = 15$. The size of a region is proportional to the scale at which it obtained its extrema value. For the sake of uniformity, all regions are scaled to a size of 41×41 pixels using bicubic interpolation before region descriptors are computed. Figure 2 shows sample interest regions extracted with the aforementioned detector.

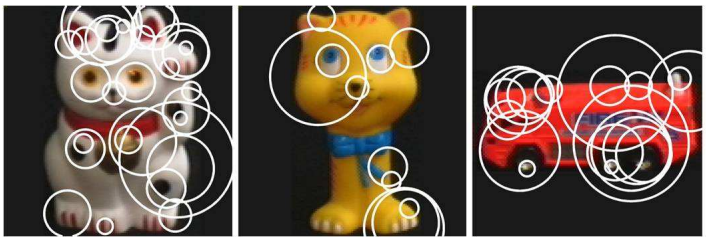


Figure 2: Detected regions on three images from the COIL-100 data set.

Features	Description
Gradient information	*Gradient*, *Gradient magnitude* and *Gradient Orientation* $(\nabla, \parallel \nabla \parallel, \nabla_\phi)$.
Gabor filter response	The sum of *Gabor filters* with 8 different orientations (gab).
Interest operators †	The response to 3 stable interest operators: *Harris*, *IPGP*1 and *IPGP*2 $(K_{Harris}, K_{IPGP1}, K_{IPGP2})$.
Color information	All the channels of 4 color spaces: *RGB*, *YIQ*, *Cie Lab*, and *rg chromaticity* (R,G,B,Y,I,Q,L,a,b,r,g).

† K_{IPGP1} is proportional to a DoG filter, and K_{IPGP2} is based on the determinant of the Hessian [9, 10].

Table 1: The complete feature space Φ.

Texture and Color Features. In order to appropriately describe each image region the search space Φ of possible features includes 18 different types of color and texture related information, see Table 1. To characterize the information contained along different channels, six statistical descriptors are computed: *mean* μ, *standard deviation* σ, *skewness* γ_1, *kurtosis* γ_2, *entropy* H and *log energy* E. This yields a total of 108 possible descriptor values for the multivariate GMMs. Because general statistical information is used, the descriptors will mostly be rotationally invariant.

Genetic Algorithms (GA) are stochastic heuristic search techniques that model, in an abstract manner, the principles of natural evolution [2]. The basic principles that a canonical GA follows are survival of the fittest (selection), recombination and replication of fit genetic material (crossover), and the introduction of novel genetic information (mutation), all of which are modeled as stochastic processes. These techniques operate over a set of parameterized solutions using population-based metaheuristics. GAs can manage a number of constraints and design decisions, and carry out a search in an intrinsic parallel manner; thence, GAs can be considered as a global optimization and search method. In the current work, the canonical GA with a binary string chromosome is employed.

Gaussian Mixture Models are a useful tool when it is necessary to model multimodal data, or as an approximation to different types of more complex distributions. The GMM pdf is defined as a weighted sum of Gaussian pdfs,

$$p(\mathbf{x};\Theta) = \sum_{c=1}^{C} \alpha_c \mathcal{N}(\mathbf{x};\mu_\mathbf{c},\Sigma_c) \;, \tag{2}$$

where $\mathcal{N}(\mathbf{x};\mu_\mathbf{c},\Sigma_c)$ is the *cth* multivariate Gaussian component with mean μ_c, covariance matrix Σ_c, and an associated weight α_c. Estimation of the mixture model parameters is

done using the EM algorithm when a fixed number of components is assumed. Alternatively, if a variable number of component is desired, with a maximum bound, it is possible to use the the Greedy-EM [7]. Classification with GMMs can be done through Bayes rule, or using confidence-based classification [7]. A confidence value $\kappa \in [0,1]$ and confidence region $\mathscr{R} \subseteq \Phi$ for a pdf are $0 \leq p(\mathbf{x}) < \infty$, $\forall\, \mathbf{x} \in \Phi$. κ is a confidence value related to a non-unique confidence region $\mathscr{R}$ such that

$$\int_{\Phi \setminus \mathscr{R}} p(\mathbf{x}) d\mathbf{x} = \kappa \,. \tag{3}$$

A sample $\mathbf{x}$ that lies within $\mathscr{R}$ is considered a true member of the class modeled by p, otherwise it is classified as an outlier.

Fisher's Linear Discriminant. Fisher defined the separation between two distributions $\mathscr{N}_i$ and $\mathscr{N}_j$ as the following ratio

$$S_{i,j} = \frac{(\mathbf{w}(\mu_i - \mu_j))^2}{(\mathbf{w}^T(\Sigma_i + \Sigma_j)(\mathbf{w}))} \,, \tag{4}$$

where $\mathbf{w} = (\Sigma_i + \Sigma_j)^{-1}(\mu_i - \mu_j)$ [1]. Note that S is defined for unimodal pdfs, hence a weighted version $\widehat{S}$ that accounts for the weight α_i and α_j of the associated Gaussian components in a GMM is proposed, such that

$$\widehat{S}_{i,j} = \frac{S_{i,j}}{1 + \alpha_i + \alpha_j} \,. \tag{5}$$

Hence, the separation between components with a small combined weight (they have less influence over their associated models) will *appear* to be larger with respect to the separation between components with larger weights. Therefore, let C_a and C_b represent the number of components of $p_a(\mathbf{x};\Theta_a)$ and $p_b(\mathbf{x};\Theta_b)$ respectively, then $S^{a,b}$ represents the *apparent* separation matrix of size $C_a \times C_b$ that contains the weighted separation $\widehat{S}_{i,j}$ of every component of p_a with respect to every component of p_b. The final *apparent* separation measure $\mathscr{S}$ between p_a and p_b is given by

$$\mathscr{S}^{a,b} = inf(S^{a,b}) \,. \tag{6}$$

3 Proposed Approach

This section describes the details of the proposed approach to object recognition, novel object detection, and salient object estimation; a flowchart view is depicted in Figure 3.

3.1 Learn Object Models

First, there is an initial off-line step in which interest regions from every object $O \in M$ are extracted and labeled accordingly; moreover, all 108 descriptor values are computed for each region. Afterwards, the GA performs feature selection, and learns appropriate GMMs for a subset N of the objects in M. Figure 3a shows the basic flow chart of a canonical GA, the two main aspects to discuss is how candidate solutions are represented and how fitness assignment is done. The other processes in the GA are standard: fitness proportional selection, mask crossover, single bit mutation and elitist survival strategy.

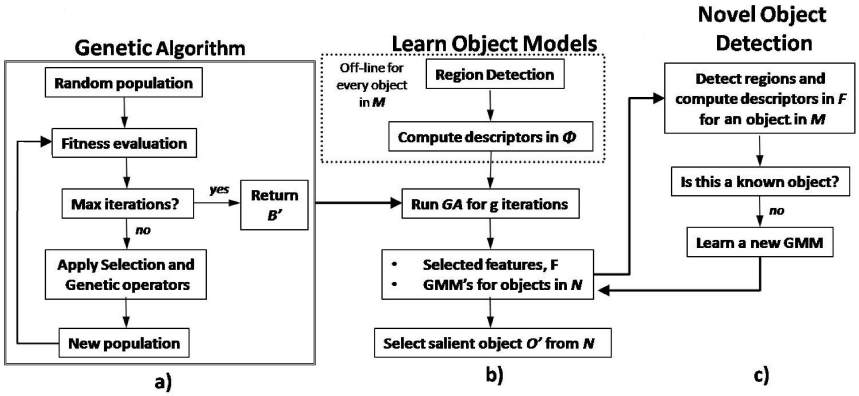


Figure 3: An overview of the proposed approach, a) Genetic Algorithm, b) Learn object models, c) Novel object detection.

Solution Representation: Each individual in the population is coded as a binary string $B = (b_1, b_2, ...b_{108})$ of 108 bits. Each bit is associated with one of the statistical descriptors in Φ. Therefore, if bit b_i is set to 1 its associated descriptor will be selected, with the opposite being true if $b_i = 0$. The feature vector $\mathbf{x}_\lambda$ for each region λ is thereby given by the concatenation of the set of selected descriptors $F \subseteq \Phi$.

Fitness Evaluation: Here is where object models are learned and fitness is assigned to each individual in the population. For every object $O_j \in N$ a corresponding GMM $p_j(\mathbf{x};\Theta_j)$ is trained with a *one vs. all* strategy with 70% of the regions, using the descriptor values selected by *B*. The GMM classifiers are trained with the EM algorithm. After training, a set $\mathscr{P} = \{p_i(\mathbf{x};\Theta_i)\}$ of $|N|$ GMMs, on each $\forall\, O_i \in N$. Afterwards, the remaining 30% of image regions are used for testing and a corresponding accuracy score $\mathscr{A}_i$ is computed using Bayes rule. Optimization is posed as a minimization problem, hence fitness is assigned by

$$f(B) = \begin{cases} \dfrac{B_{ones}+1}{\mathscr{A}' \cdot inf(\mathscr{S}^{p_i,p_j})} & \forall\, p_i, p_j \in \mathscr{P}\ , i \neq j\ , \quad when \quad \forall\, \mathscr{A}_i > 0\ , \\ \dfrac{K \cdot B_{ones}+1}{\mathscr{A}'+\varepsilon} & otherwise\ . \end{cases} \tag{7}$$

In the above equation, B_{ones} is the number of ones in string *B*, $\mathscr{A}'$ is the average accuracy score of all the GMMs in $\mathscr{P}$, a penalization term set to $K = 2$, and $\varepsilon = 0.01$; hence, fitness depends upon testing and not training accuracy. The first case in Eq. 7 is applied when all of the classifiers where able to obtain an accuracy score, fitter individuals will minimize the number of selected descriptors and maximize the average testing accuracy $\mathscr{A}'$. Furthermore, the term $inf(\mathscr{S}^{p_i,p_j})$ promotes between-class model separation by selecting the infimum of all the apparent separation measures computed for every object in *N*. On the other hand, the second case in Eq. 7 is applied when the EM algorithm fails to produce a valid GMM for one of the objects in *N*.

After a fixed number of iterations the GA stops and returns the fittest individual B^o

found so far. The best individual B^o is re-trained using the Greedy-EM instead of the basic EM, this is done for two reasons. First, the Greedy EM did not prove to be appropriate during evolution because it required more computation time and produced more runs that failed to converge. Secondly, once the GA has produced a valid high performance solution, the associated object models can be further enhanced by using the Greedy EM on B^o. Therefore, the GA returns the selected subset of descriptors F that characterize the objects in N, and a set of trained GMMs $\mathscr{P}^o$. Finally, the most salient object O^o in N is said to be modeled by the GMM p^o that satisfies the following,

$$p^o \leftarrow arg \max_{p_i}(\mathscr{S}^{p_i,p_j}) \quad \forall\, p_i, p_j \in \mathscr{P}^o \quad with \quad i \neq j\ . \tag{8}$$

3.2 Object Recognition and Novel Object Detection

In order to test the ability of the described approach to recognize known objects and detect novel objects (those without a corresponding $p_i \in \mathscr{P}^o$) the process in Figure 3c is followed. Given an image of an object $O_i \in M$, interest regions are detected and their corresponding descriptors, specified in F, are computed. The extracted regions are classified using confidence estimation with the models in $\mathscr{P}^o$. A confidence region within each GMM in $\mathscr{P}^o$ is defined, with the confidence threshold set to $\kappa = 0.95$. Therefore, if a large majority, over 60%, of the regions lie within the confidence region of a given $p_j \in \mathscr{P}^o$ then it is said that object $O_i = O_j$, thereby accounting for a successful recognition. Otherwise, if regions are classified as outliers from all known classes, it is possible to tag them as belonging to an object not modeled in $\mathscr{P}^o$. Hence, if the percentage of regions classified as outliers is $\mathscr{A}_{out} > 60\%$, then the sampled object O_i is labeled as a new object, and a corresponding GMM is learned and added to $\mathscr{P}^o$.

4 Experimental Results

This section presents three different experiments to test the proposed object recognition system. The code was written mostly in MATLAB, the GMMBAYES Toolbox[1] was used for GMM training, and the Genetic Algorithms for Optimization Toolbox[2] was used as part of the GA code. The images used for testing are taken from the COIL-100 data set, Figure 4 shows the first 40 objects in the data set [6]. Every object is seen from 72 different views, interest regions are extracted from all of the views and tagged accordingly as the ground truth for each object. The basic parameters of the algorithm are the same in every run, only modifying the number of different objects used, the size of sets (M,N). Three experiments are presented: Exp. 1 (10,5) with objects 1 - 10 from the data set; Exp. 2 (20,10) with objects 20 - 40; and Exp. 3 (40,25) with objects 1 - 40. The GMM classifiers were trained using EM with one Gaussian component, and if a solution was not found, the algorithm is restarted with 2 components, and so on. The results presented for each experiment are shown for object recognition and novel object detection. Table 2 shows the average accuracy score obtained after the initial object models are generated (Figure 3b), along with the fitness value, the number of features, the set of selected features F, errors in object recognition, and the salient object within the set. Table 3 presents the accuracy

[1] GMMBAYES Matlab Toolbox http://www.it.lut/project/gmmbayes

[2] Genetic Algorithms for Optimization Toolbox by Andrey Popov http://automatics.hit.bg

Figure 4: These are the first 40 objects in the COIL-100 data set used in the reported experimental runs. The images used with the first two experiments are marked, while all 40 are used in the third. Salient objects selected by the separation criterion are circled. Object 32 is the only one for which novel object detection failed with $h = 60\%$.

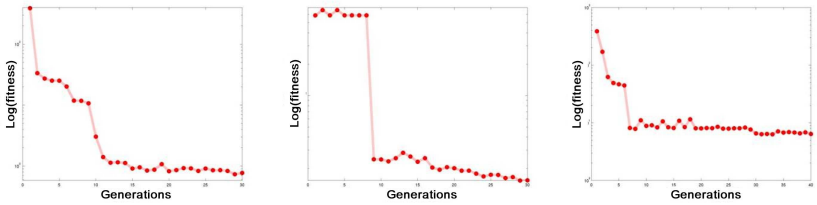
Exp.	$\mathscr{A}'$	$\mathbf{f(B^o)}$	$\mathbf{B^o_{ones}}$	**Features**	**Error**	$\mathbf{O^o}$
1)	99.6	0.5	27	$\nabla_{(\gamma_2,H)}$, $\parallel \nabla \parallel_{(\sigma,\gamma_2)}$, $\nabla_{\phi\,(\gamma_2)}$, $K_{Harris(E)}$, $K_{IPGP1(\sigma)}$, $R_{(\mu,H)}$, $G_{(\sigma,\gamma_1)}$, $B_{(\mu,\sigma,\gamma_1,H)}$, $Y_{(\mu,\gamma_2,H,E)}$, $I_{(\sigma,H)}$, $L_{(\sigma,E)}$, $a_{(\mu,\sigma)}$, $b_{(\sigma,E)}$, $g_{(\mu)}$	none	4
2)	99.2	1.5	43	$\nabla_{(\mu,\sigma,\gamma_2,E)}$, $\parallel \nabla \parallel_{(\sigma,\gamma_2,H)}$, $\nabla_{\phi\,(\mu,\sigma,\gamma_2)}$ $K_{Harris(\gamma_1,H)}$, $K_{IPGP1(\mu,E)}$, $K_{IPGP2(\gamma_1,E)}$, $gab_{(\gamma_1)}$, $R_{(\mu,\sigma,E)}$,$G_{(\mu)}$,$B_{(\sigma,\gamma_1,\gamma_2,H)}$, $Y_{(\mu,\gamma_2,H,E)}$, $I_{(\sigma,H)}$, $Q_{(\gamma_2,H)}$, $L_{(\mu)}$,$a_{(\mu,\sigma,H)}$, $b_{(\mu,\sigma,E)}$, $r_{(\mu,\sigma)}$, $g_{(E)}$	none	25
3)	98.7	6.4	37	$\nabla_{(\mu,\sigma,\gamma_2)}$, $\parallel \nabla \parallel_{(\mu,\sigma,\gamma_1,\gamma_2,H,E)}$, $\nabla_{\phi\,(\gamma_1,\gamma_2,H)}$, $K_{Harris(\gamma_2,E)}$,$K_{IPGP1(H)}$, $K_{IPGP2(\gamma_2,H)}$, $gab_{(\mu)}$, $R_{(\mu,\gamma_1,E)}$, $G_{(\mu)}$, $B_{(\gamma_2,H)}$, $Y_{(\mu,\sigma,\gamma_2,H,E)}$, $I_{(E)}$, $Q_{(\mu,\gamma_1)}$, $a_{(\gamma_1)}$, $b_{(\sigma)}$, $r_{(\sigma,H)}$, $g_{(\mu,\sigma)}$	none	4

Table 2: Performance when initial class models are learned; see text for further details.

	$\mathscr{A}'_{\mathbf{M}}$	**Errors**	**Salient Objects**
Exp.1	99.72	none	objects 4, 7, 3
Exp.2	99.04	object 32	objects 36, 38, 25
Exp.3	98.68	object 32	objects 36, 28, 4

Table 3: Performance for novel object detection. Note that $\mathscr{A}'_{\mathbf{M}}$ represents the accuracy of region classification after a corresponding model is learned for every object $O \in M$.

score once a corresponding model is learned for every object $O \in M$, the incorrectly classified objects, and the three most salient objects found in each case. Given the high level of accuracy in both sets of results, in can be concluded that the problem of object recognition is almost perfectly solved for the set of images employed. Figure 5 shows the convergence graphs of each GA run, plotting the fitness of the best individual B^o found so far. The experiments were executed with 30, 30 and 40 iterations respectively.



Experiment 1 Experiment 2 Experiment 3

Figure 5: Convergence plots that show the $log(f(B^o))$ of the best individual found thus far by the GA in each of the experimental runs.

5 Discussion and Conclusions

The results presented in the previous section exhibit promising performance patterns. For all three experiments the algorithm was able to train extremely accurate classifiers using a fraction of the available descriptors. It is important to note that in Table 2 even do all experiments produce similar values for accuracy and number of descriptors, their associated fitness scores are different. This is due to the model separation measure $inf(\mathscr{S}^{p_i,p_j})$ in the fitness function, because with more objects the space of possible objects models becomes crowded. All the classifiers trained in each experiment finished with a single Gaussian component, an unexpected outcome that can nevertheless be explained. Every object is small and tends to exhibit regular patterns across their surface; therefore, it was possible to characterize them with a single component in feature space. This suggests that GMMs would be more appropriate dealing with images that have a larger variations in descriptor space. Additionally, the convergence graphs in Figure 5 show two different patterns. First, starting from the random population the initial iterations produce very poor results, individuals in these generations are evaluated using the second case of the fitness function because the EM fails to find a valid model for at least one of the objects. Therefore, initial iterations attempt to find solutions B that are able to produce a classifier for every object in N. Once a good solution is found, and its genetic material begins to propagate throughout the population, the GA begins to optimize using the first case of the fitness function. With a valid classifier for every object it is then possible for the GA to explore the pruning of the feature space. Regarding novel object detection, the approach produced nearly perfect results with only one false negative, object 32. However, object 32 is almost identical to object 29, they only differ slightly in color space. Perhaps an interest operator that uses color information explicitly could help avoid ambiguous situations such as this. Finally, regarding the estimation of the most salient object within a

set, the algorithm also produced coherent selections. The objects selected as most salient, shown in Figure 4, are appreciably different than the rest, these objects tend to lack texture and exhibit small color variations. Furthermore, all of the other objects in the data set tend to have at least one similar counterpart, i.e. more that one toy car, and various small boxes. In conclusion, the proposed approach produced promising initial results for object recognition, novel object detection and salient object estimation. Future work concentrates on using images with complex backgrounds, in order to perform scene classification of real world images where the benefits of a multimodal model are expected to become evident.

Acknowledgements

Research funded by UC MEXUS-CONACyT Collaborative Research Grant 2005 through the project "Intelligent Robots for the Exploration of Dynamic Environments", the Ministerio de Educación y Ciencia through the project Oplink - TIN2005-08818-C04, and Junta de Extremadura Spain. First author supported by scholarship 174785 from CONACyT México. This research was also supported by the LAFMI project, and special thanks is given to the Complex Team - INRIA Roquencourt and Grupo de Evolución Artificial - Universidad de Extremadura campus Mérida.

References

[1] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

[2] John H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, Cambridge, MA, USA, 1992.

[3] David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision, 20-25 September, 1999, Kerkyra, Corfu, Greece*, volume 2, pages 1150–1157. IEEE Computer Society, 1999.

[4] M. Markou and S. Singh. A neural network-based novelty detector for image sequence analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(10):1664–1677, 2006.

[5] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(10):1615–1630, 2005.

[6] S.A. Nene, Nayar, and H. S.K., Murase. Columbia object image library (coil-100). Technical report, Department of Comuter Science. Columbia University, 1996.

[7] Pekka Paalanen, Joni-Kristian Kamarainen, Jarmo Ilonen, and Heikki Kälviäinen. Feature representation and discrimination based on gaussian mixture model probability densities-practices and algorithms. *Pattern Recognition*, 39(7):1346–1358, 2006.

[8] Cordelia Schmid and Roger Mohr. Local grayvalue invariants for image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(5):530–534, May 1997.

[9] Leonardo Trujillo and Gustavo Olague. Synthesis of interest point detectors through genetic programming. In Mike Cattolico, editor, *Proceedings of GECCO 2006*, volume 1, pages 887–894. ACM, 2006.

[10] Leonardo Trujillo and Gustavo Olague. Scale invariance for evolved interest operators. In Mario Giacobini et al., editor, *Proceedings of EvoWorkshops 2007*, volume 4448 of *Lecture Notes in Computer Science*, pages 423–430. Springer, 2007.

[11] Leonardo Trujillo, Gustavo Olague, Pierrick Legrand, and Evelyne Lutton. Regularity based descriptor computed from local image oscillations. *Optics Express*, 15:6140–6145, 2007.

Navier-Stokes formulation for modelling turbulent optical flow

Ashish Doshi and Adrian G. Bors

Dept. of Computer Science, University of York, York YO10 5DD, UK
{adoshi,adrian.bors}@cs.york.ac.uk

Abstract

This paper proposes a physics-based methodology for the analysis of optical flows displaying complex patterns. Turbulent motion, such as that exhibited by fluid substances, can be modelled using fluid dynamics principles. Together with supplemental equations, such as the conservation of mass, and well formulated boundary conditions, the Navier-Stokes equations can be used to model complex fluid motion estimated from image sequences. In this paper, we propose to use a robust kernel which adapts to the local data geometry in the diffusion stage of the Navier-Stokes formulation. The proposed kernel is Gaussian and embeds the Hessian of the local data as its covariance matrix. The local Hessian models the variation of the flow in a certain neighbourhood. Moreover, we use a robust statistics mechanism in order to eliminate the outliers from the estimation process. The proposed methodology is applied on artificial vector fields and in image sequences showing atmospheric and solar phenomena.

1 Introduction

Classical optical flow estimation methods work on the assumption that image intensity structures are approximately constant under motion [1, 8]. Robust estimation employing either median statistics or diffusion has been used to eliminate outliers from the optical flow [4] and to smooth colour images while preserving edges [3], respectively. Recently, robust statistics and diffusion have been embedded in a smoothing kernel for jointly processing the data statistics and the local geometry in noisy optical flows [6]. This method was shown to preserve data characteristics as well as the boundaries of the moving objects, while resulting in smoothed optical flows.

Very often, the natural phenomena modelling involve the motion of dynamic fluids which differs radically from that of rigid bodies. Classical optical flow estimation algorithms would fail in such cases. The use of fluid flow modelling for motion estimation can be traced back to the work of Fitzpatrick [7], who compared optical and fluid flow methods. The computation of flows depends largely on the specific nature of the application. Using Fitzpatrick's analysis as a basis, Song and Leahy [12], employed the equation of continuity as an additional constraint to Horn and Schunck's algorithm [8] in order to obtain better motion estimation of the beating heart. Navier-Stokes equations

have been extensively studied in fluid mechanics for modelling the behaviour of fluids under various conditions and constraints [9]. The Navier-Stokes and optical flow constraint equations have been employed for modelling Karman flows in [10]. Bertalmio *et. al.* applied the Navier-Stokes equations to image and video inpainting [2]. Their approach uses the vorticity-stream formulation of the fluid flow equation, which can be attributed to the image intensity-Laplacian relationship. Corpetti *et. al.* used the vorticity-stream formulation to recover dense motion of water vapours [5].

Navier-Stokes equations have been used in computer graphics for visualising flames and building animation tools based on fluid-like motion [11, 13, 14]. The stable fluid solver (SFS) algorithm implements Navier-Stokes equations and consists of a set of consecutive processing steps [13], such as: advection, diffusion and mass conservation. The boundary conditions are important in constraining the fluid motion [9]. The boundaries have been processed as a set of constraints on a grid [14], by enforcing repetition and employing the Fast Fourier Transform (FFT) [13] or by using level sets [11]. In this study, we extend the SFS solver methodology and apply it for smoothing vector fields estimated from image sequences representing turbulent moving fluids. In our approach, the diffusion step is anisotropic and robust by considering a median of the Hessian diffusion kernel [6]. The proposed hybrid SFS method processes the local geometry and data statistics consistently with the flow motion. The proposed approach is applied for smoothing artificial vector fields and in two image sequences. The paper is structured as follows: Section 2 outlines the SFS algorithm, while Section 3 describes our hybrid solver applied for modelling vector fields. Experimental results and their analysis are presented in Section 4, while Section 5 concludes the paper.

2 The Stable Fluid Method

Navier-Stokes methodology represent the basis for modelling a large variety of phenomena such as those characterising weather, ocean currents, water flow in a pipe, the air flow around a wing, the motion of stars inside a galaxy, blood flow, economics behaviour, etc [9]. In engineering, they are used in the analysis of the effects of pollution, the design of aircraft and of power stations, etc. Navier-Stokes methodology has been applied in Computer Graphics in order to visualise and create the effects given by the complex movement of fluids such as that of coloured gases, air, clouds, liquids, smoke, fire, etc., [11, 13]. The explicit model is generally used for precise computation of fluid dynamics and involves heavy computational complexity [9]. The Von Neumann's stability analysis, as shown in [9], highlights that the implicit model of discretisation when calculating Navier-Stokes equations is unconditionally stable, although it requires a complex numerical implementation scheme. The SFS algorithm proposed by Stam represents an implementation of the Navier-Stokes methodology in an implicit scheme [13, 14].

In order to achieve visual effects, the Navier-Stokes equations are used for both density and velocity in the SFS algorithm [13, 14]. Unlike in the original SFS approach, in this study we consider only the modelling of motion based on the Navier-Stokes equations. The area of investigation (in our case an image or a segmented region from an image) is split into cells located on a grid and we associate a particle to each grid location. Let us assume that the SFS system moves the particles around according to a vector field, where each vector corresponds to a grid location. The Navier-Stokes equation for a given system is derived using the conservation of mass, momentum, and energy for an

arbitrary control volume [9] and is given by :

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u}\cdot\nabla)\,\mathbf{u} - \frac{\nabla P}{\rho} + \nu\nabla^2\mathbf{u} + \mathbf{f} \tag{1}$$

where the change of velocity $\mathbf{u}$ over time is represented with respect to the advection, gradient of the pressure P, diffusion and external forcing function $\mathbf{f}$, while ν is a viscosity constant that characterises the fluid and ρ is a parameter. The pressure is assumed to be constant in the given field and its gradient is zero, *i.e.* the change in pressure from one spatial position to another in the vector field is negligible. Consequently, the equation employed by the SFS method is :

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u}\cdot\nabla)\,\mathbf{u} + \nu\nabla^2\mathbf{u} + \mathbf{f} \tag{2}$$

The diffusion term $\nu\nabla^2\mathbf{u}$ characterises fluids which are assumed incompressible and Newtonian. Moreover, for incompressible fluids it is important to enforce the conservation of mass [9]:

$$\nabla\cdot\mathbf{u} = 0 \tag{3}$$

which states that the divergence of velocity components is zero for infinitesimal time steps. The density of a particle is constant between iterations, thereby the total mass of the field is conserved within the given region.

```
    for k ← 1 to ▷ convergence / number of iterations
        do
1           add force: u_1 = u_0 + f Δt
2           advect: u_2(x) = advect(u_1(x, −Δt))
3           transform: û_2 = FFT(u_2)
4           diffuse: û_3(z) = û_2(z)/(1 + νΔtk^2)
5           conserve: û_4 = conserve(û_3)
6           transform: u_4 = FFT^{-1}(û_4)
```

Figure 1: The stable fluid solver algorithm.

The SFS algorithm proceeds to calculate the velocity components $\mathbf{u}$ as described in Fig. 1, [13]. For each iteration, the first step consists of adding the external forcing function $\mathbf{f}$ which determines the initial conditions in the processing cycle. The second step represents the advection term in equation (2), which corresponds to the following :

$$(\mathbf{u}\cdot\nabla)\,\mathbf{u} = \left(u_x\frac{\partial u_x}{\partial x} + u_y\frac{\partial u_x}{\partial y}, u_x\frac{\partial u_y}{\partial x} + u_y\frac{\partial u_y}{\partial y}\right) \tag{4}$$

where $\mathbf{u} = (u_x, u_y)$. The analysis of the advection process in real physical phenomena is provided in [9]. The process described by equation (4) is known as the self-advection of velocity. The advection step from the SFS algorithm is implemented by moving the motion vector of each grid cell back in time with $-\Delta t$ by backtracking the velocity field. The third step transforms the velocity field to the frequency domain using the Fast Fourier Transform (FFT). The requirement to set specific boundary conditions is eliminated by extending the spatial repeatability of the area under consideration and by applying FFT. The diffusion term (fourth step) represents the decay of high spatial frequencies in the velocity field and is computed in the Fourier domain with a Gaussian filter processing the velocity component $\mathbf{u}$ by using the time step Δt and the fluid kinematic viscosity ν. The finite difference implicit scheme is used here to discretise the diffusion term in

order to obtain an unconditionally stable system [13]. The fifth step enforces the local incompressibility of the optical flow which requires that the amount of flow entering in a specific area should be equal with the flow exiting that area. The final step projects the flow back from the frequency domain to the spatial-time domain using the inverse FFT transform. This algorithm was modified in [14] by replacing the FFT transformations and the processing in the frequency domain with defining a set of boundary constraints on a grid-based representation of the flow.

3 The Robust Hybrid Fluid Solver

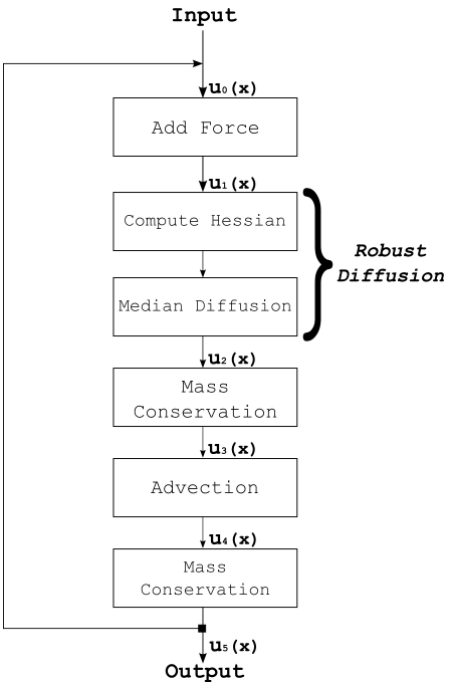


Figure 2: Robust hybrid solver.

The implementation of the stable fluid solver [13] provided rather poor performance in modelling turbulent optical flow estimated from image sequences. This is mainly caused due to the uncertainty in the initial estimation of the optical flow which leads to noise, particularly in image sequences displaying complex motion. In order to improve the performance on optical flow, we propose to embed a robust anisotropic kernel [6] in the diffusion step of the SFS. Fig. 2 shows a flow diagram of the proposed robust hybrid fluid solver. The initial flow can be estimated using the block matching algorithm as in [4] or other motion estimation algorithms [1]. Optical flows provided by block-matching or by using temporal gradient estimation are invariably noisy [4], particularly in the case of image sequences representing moving fluids or other complex phenomena.

The first processing block corresponds to a reinforcement step and in the proposed method is implemented by adding a proportion of the velocity from the previous iteration

to the current velocity :

$$\mathbf{u}_1(t+\Delta t) = (1-\varepsilon)\mathbf{u}_0(t) + \varepsilon\Delta t\mathbf{u}_5(t) \tag{5}$$

where $\mathbf{u}_5(t)$ is the motion vector from the previous iteration t, $\varepsilon \in (0,1)$ is a weighting factor modelling the degree of the reinforcement and $\mathbf{u}_0(t), \mathbf{u}_1(t+\Delta t)$ represent the motion vector reinforced by force at times t and $t+\Delta t$, respectively. At the first iteration there is no reinforcement, *i.e.* $\varepsilon = 0$. The SFS algorithm described in Section 2 proposes to advect the initial flow at Step 2 from Fig. 1. However, that algorithm produces unreliable estimation when applied to noisy vector fields. The optical flow should have a degree of smoothing before advection can be applied. In our approach, we propose to diffuse the noisy flow before proceeding to the advection stage. The transfer function of the original smoothing algorithm is a Gaussian function appropriately defined within the frequency domain [13]. In our approach, we propose to implement a Hessian based diffusion that jointly processes the local geometry and the statistics of the local vector field as in [6] :

$$\hat{\mathbf{u}}_2(t+\Delta t) = \frac{\sum\limits_{\mathbf{x}_i \in \eta(\mathbf{z}_c)} \mathbf{u}_{1,i}(t) \exp[-(\mathbf{x}_i - \mathbf{z}_c)^T \mathbf{H}^{-1} (\mathbf{x}_i - \mathbf{z}_c)]}{\sum\limits_{\mathbf{x}_i \in \eta(\mathbf{z}_c)} \exp[-(\mathbf{x}_i - \mathbf{z}_c)^T \mathbf{H}^{-1} (\mathbf{x}_i - \mathbf{z}_c)]} \tag{6}$$

where $\hat{\mathbf{u}}_2(t+\Delta t)$ is the intermediate diffused value, $\mathbf{H}$ represents the local Hessian, $\mathbf{u}_{1,i}(t)$ is the vector at location i within a neighbourhood $\eta(\mathbf{z}_c)$, centred at the location $\mathbf{z}_c$. The Hessian of the optical flow is calculated locally as :

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 \mathbf{u}}{\partial x^2} & \frac{\partial^2 \mathbf{u}}{\partial x \partial y} \\ \frac{\partial^2 \mathbf{u}}{\partial y \partial x} & \frac{\partial^2 \mathbf{u}}{\partial y^2} \end{bmatrix} \tag{7}$$

The eigenvector corresponding to the largest eigenvalue shows the local direction of the optical flow. This diffusion kernel is anisotropic and adapts to the local structure of the optical flow. Significant optical flow transitions are detected and consequently not smoothed over by the Hessian-based kernel. However, anisotropic diffusion does not deal properly with outliers as shown in a study provided in [6]. In order to properly process the local statistics and eliminate outliers, the median algorithm is considered for robustifying the Hessian based diffusion in the neighbourhood $\eta(\mathbf{z}_c)$.

At the advection stage, our model is only concerned with the nonlinearity of the advection term from equation (4). As mentioned in the previous Section, the self-advection term represents the ability of the velocity components to move their own values from one position to another on a grid in a time step interval, Δt. This procedure involves interpolating the velocity at the grid points, using a neighbourhood approximation, from the previous time step back to the position in the current time step [14].

The model is dependent on the initialisation and on boundary conditions of the system under study. Boundary condition are specifically provided onto the grid in order to represent the physical limits of the optical flow. Such boundary conditions can be the result of image or motion segmentation algorithms or of *a priori* information about the image sequence. There are two boundary conditions to consider. The first condition is determined by the physical boundary. This is represented by the Von Neumann condition which specifies the normal component of the flow to the boundary surface as :

$$\left.\frac{\partial \mathbf{u}}{\partial \mathbf{n}}\right|_{\Omega} = 0 \tag{8}$$

where Ω represents the boundary and $\mathbf{n}$ is its surface normal. This means that the wall absorbs any flow particles coming towards it. For the sake of reducing the required computation complexity, the walls of the domain, Ω are represented by zero values on a geometric grid, which are enforced at every stage of the computation in order to preserve the stability and integrity of the numerical calculation. Since our proposal incorporates both explicit and implicit finite differencing schemes, it is absolutely imperative that the model adheres to the stability criteria, given by $\Delta t/(\Delta \mathbf{x})^2 \leq 1/2$, where $\Delta \mathbf{x}$ represents the location change during the time interval Δt.

The second condition relates to the conservation of mass of the velocity field. The conservation of mass, given by equation (3), should be maintained in order to ensure the incompressibility of the flow. In order to maintain a divergence free velocity field for every stage of computation, the conservation of mass is enforced after both diffusion and advection stages. The conservation of mass stage corresponds to a data normalisation process. The conservation of mass is enforced by using the Helmholtz-Hodge decomposition [13] of the velocity field. This decomposition provides an exact solution so that the mass conserved incompressible flow can be obtained by extracting the gradient of the flow from the current vector field. This decomposition maintains the incompressibility and smoothness of the estimated velocity field. Mass conservation is important for realistically estimating optical flow of fluids. For exemplification, the Helmholtz-Hodge decomposition of the exact closed cavity laminar flow (artificial data experiment provided in Section 4) at the 1000th iteration is shown in Fig. 3.

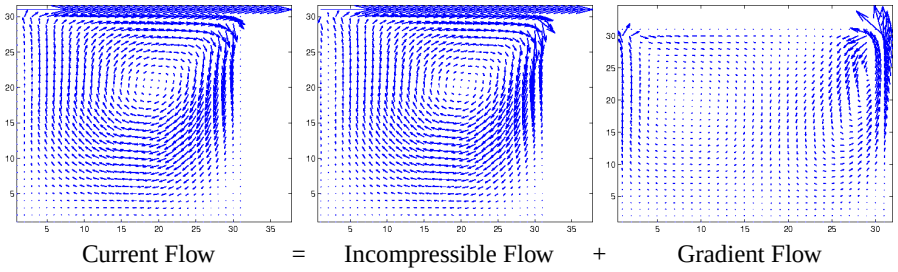
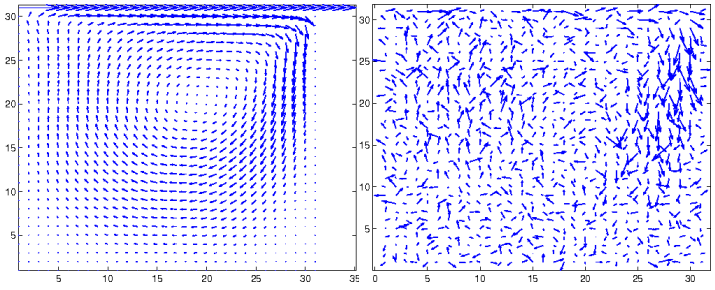


Figure 3: Helmholtz-Hodge decomposition of a closed lid driven cavity laminar flow.

4 Experimental Results

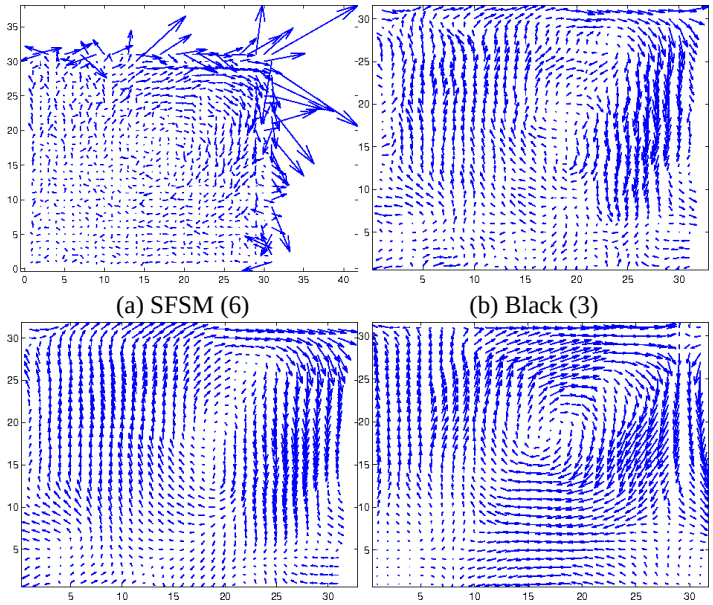
We present results when the proposed algorithm is evaluated on a synthetic vector field and on the optical flow estimated from two real-world image sequences. The synthetic sequence is created using the original Navier-Stokes equations [9] depicting the air flow generated within a lid driven closed cavity. The synthetic flow is created using the vorticity-stream formulation of the Navier-Stokes equations instead of the classic velocity-pressure formulation. Fig. 4(a) represents the simulated synthetic field that visualises the air flow moving with a fixed velocity from left to right inside the top area of a closed cavity. This flow has been obtained after applying the Navier-Stokes equation for a thousand iterations. Fig. 4(b) shows flow degradation after adding Gaussian noise with zero mean and variance $\sigma^2 = 0.25$. Modelling results using the modified SFS (SFSM) algorithm [14] adapted for usage on vector fields is shown in Fig. 5(a), while vector field smoothing using Black's anisotropic diffusion algorithm [3] is provided in Fig. 5(b). Fig. 5(c) shows

the effects of using MED-2DH which is a robust Hessian based diffusion algorithm described in [6], while the robust hybrid fluid solver embedding the median of 2D Hessian diffusion kernel (MedH-SFS) algorithm, as described in Section 3, is shown in Fig. 5(d).



(a) Ground truth synthetic flow (b) $\sigma^2 = 0.25$

Figure 4: Synthetic closed lid-driven cavity flows



(a) SFSM (6) (b) Black (3)

(c) MED-2DH (4) (d) MedH-SFS (5)

Figure 5: Artificial vector field smoothing comparisons. For better visualisation, the vector from the upper-right corner of the SFSM vector field in (a) has been rescaled.

The results in Fig. 5 are obtained at convergence when the mean square error difference between vector fields at two successive iterations is less than 0.01. The number of iterations necessary to achieve convergence is provided in the parentheses from the caption of each result plot of Fig. 5. From these results, we can observe that the vector field modelled by SFSM is still noisy at convergence, while the noise has been significantly reduced in the other smoothed vector fields. It can be observed that MedH-SFS provides the best results and the flow vortex recovered is better located when compared to the vortices recovered using Black and MED-2DH.

Gaussian Noise (σ^2)	SFSM	SFS	MedH-SFS	Black	MED-2DH
0.01	0.7525	0.6211	**0.7634**	0.7226	0.7383
0.10	0.6020	0.5616	**0.7327**	0.6554	0.6997
0.25	0.4538	0.4523	**0.6849**	0.5584	0.6424
0.30	0.4373	0.4624	**0.6704**	0.5567	0.6058
0.40	0.4005	0.4184	**0.5799**	0.4958	0.5556

Table 1: Mean cosine error (MCE) of smoothed vector fields.

For numerical comparisons, we consider the mean cosine error (MCE) between the recovered smoothed flow and the ground truth flow. The MCE is calculated as:

$$\text{MCE} = \frac{\sum_{i=1}^{L} \mathbf{u}_i \cdot \hat{\mathbf{u}}_i}{\|\mathbf{u}_i\| \, \|\hat{\mathbf{u}}_i\| \, L} = \frac{\cos(\theta_i)}{L} \tag{9}$$

where L is the total number of vectors, $\mathbf{u}_i$ is the ground truth before considering the noise and smoothing, and $\hat{\mathbf{u}}_i$ is the result achieved after smoothing the noisy vector field at location i. The MCE is the normalised dot product between two vectors which provides the cosine of the angle between them, denoted as θ_i. The closer MCE is to 1.0, the more similar are the two vector fields. The MCE results are provided in Table 1 after one iteration of smoothing. SFS algorithm was described in Section 2 and was adapted from [13], while SFSM was described in [14]. Both these algorithms have been adapted to work on vector fields. It can be observed that SFS provides good results for a vector field corrupted with low noise variance. However, its performance deteriorates significantly when the noise increases, because the corrupted vector field departs significantly from the Navier-Stokes underlying model. The robust diffusion hybrid fluid algorithm MedH-SFS provides better results than either SFS or SFSM methods in terms of MCE when considering additive Gaussian noise as it can be observed from Table 1. MedH-SFS is also consistently better than Black [3] and MED-2DH [6] anisotropic smoothers.

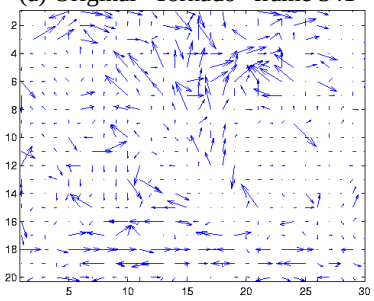
We have applied the proposed methodology on optical flows estimated from image sequences. Fig. 6(a) represents a frame from "Tornado" image sequence, while Fig. 6(b) shows a frame from the "Solar Flare" sequence obtained from Kanzelhöhe Obervatory's solar and environmental research website. The first sequence represents a complex atmospheric phenomenon while the second image is used to observe and analyse solar surface activity. The initial optical flows have been estimated using block matching algorithm (BMA) and are shown in Fig. 6(c) and Fig. 6(d), respectively. The complexity of the motion in the scenes as well as the compression artefacts influence negatively the performance of the BMA algorithm. Fig. 6(e) and Fig. 6(f) show the smoothing result when using MedH-SFS algorithm on the optical flow estimated from the "Tornado" sequence and from the "Solar Flare" optical flow, respectively, both after one iteration. The improvements provided by the Med-SFS over the initial optical flows are significant. We can clearly identify the moving twister and its boundaries after using the proposed methodology as it can be observed in the optical flow from Fig. 6(e). Turbulent movements of the solar surface can be properly identified in Fig. 6(f).

5 Conclusions

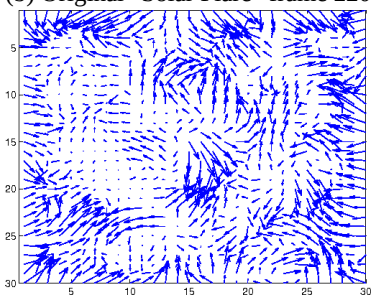
We have presented a physics based model that smoothes and models optical flow representations estimated from images representing complex and turbulent fluid motion. The

(a) Original “Tornado” frame 341

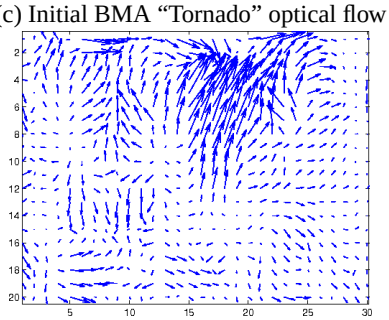
(b) Original “Solar Flare” frame 220



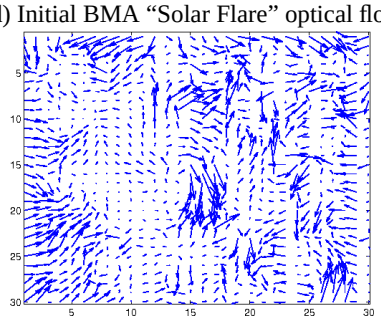
(c) Initial BMA “Tornado” optical flow



(d) Initial BMA “Solar Flare” optical flow



(e) MedH-SFS smoothed “Tornado” flow



(f) MedH-SFS smoothed “Solar Flare” flow

Figure 6: Smoothing optical flows in image sequences displaying turbulent motion.

Stable Fluid Solver (SFS) model is based on the Navier-Stokes equations for incompressible fluid. The SFS algorithm, originally developed in computer graphics for visualising fluid like movement and for building animation tools, has been modified in order to be used on optical flows. The proposed model is highly efficient and stable under certain conditions. The flow incompressibility condition is achieved by imposing the mass conservation through the Helmholtz-Hodge decomposition. We embed a robust Hessian based kernel in the diffusion step of the Navier-Stokes formulation in order to improve the performance of the proposed method for smoothing vector fields. This kernel ensures that smoothing occurs along the structure of the motion field while maintaining the general optical flow structure and the main optical flow features. The proposed kernel ensures robust statistics capability in order to reduce the impact of outliers and thus to enhance the smoothness of the resulting optical flow. The new model is shown to provide good results

in both artificial data and in optical flow from two image sequences, showing turbulent atmospheric and solar activity phenomena.

References

[1] J. L. Barron, D. J. Fleet, and S. Beauchemin. Performance of optical flow techniques. *Int. Journal of Computer Vision*, 12(1):43–77, 1994.

[2] M. Bertalmio, A. L. Bertozzi, and G. Sapiro. Navier-Stokes, fluid dynamics, and image and video inpainting. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 355–362, Kauai, HI, 2001.

[3] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger. Robust anisotropic diffusion. *IEEE Trans. on Image Processing*, 7(3):421–432, 1998.

[4] A. G. Bors and I. Pitas. Optical flow estimation and moving object segmentation based on median radial basis function network. *IEEE Trans. on Image Processing*, 7(5):693–702, 1998.

[5] T. Corpetti, *É*. Mémin, and P. Pérez. Dense estimation of fluid flows. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(3):365–380, March 2002.

[6] A. Doshi and A. G. Bors. Robust diffusion kernels for optical flow smoothing. In *Proc. IEEE Workshop on Machine Learning for Signal Processing*, pages 415–420, Maynooth, Ireland, 2006.

[7] J. M. Fitzpatrick. A method for calculating velocity in time dependent images based on the continuity equation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 78–81, San Francisco, CA, 1985.

[8] B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[9] J. D. Anderson Jr. *Computational Fluid Dynamics: The Basics with Applications*. McGraw Hill, 1995.

[10] Y. Nakajima, H. Inomata, H. Nogawa, Y. Sato, S. Tamura, K. Okazaki, and S. Torii. Physics-based flow estimation of fluids. *Pattern Recognition*, 36(5):1203–1212, 2003.

[11] S. Premoze, T. Tasdizen, J. Bigler, A. Lefohn, and R. T. Whitaker. Particle-based simulation of fluids. In *Proc. EUROGRAPHICS, Computer Graphics Forum 22(3)*, pages 401–410, 2003.

[12] S. M. Song and R. M. Leahy. Computation of 3D velocity fields from 3-D cine CT images of a human heart. *IEEE Trans. Medical Imaging*, 10(1):295–306, 1991.

[13] J. Stam. A simple fluid solver based on the FFT. *Journal of Graphics Tools*, 6(2):43–52, 2001.

[14] J. Stam. Real-time fluid dynamics for games. In *Proc. Game Developer Conference*, volume 4, pages 76–92, Mar. 2003.

All Pairs Shortest Path Formulation for Multiple Object Tracking with Application to Tennis Video Analysis

F.Yan W.Christmas J.Kittler
Centre for Vision, Speech and Signal Processing
University of Surrey
Guildford, GU2 7XH, UK
{f.yan,w.christmas,j.kittler}@surrey.ac.uk

Abstract

In previous work, we developed a novel data association algorithm with graph-theoretic formulation, and used it to track a tennis ball in broadcast tennis video. However, the track initiation/termination was not automatic, and it could not deal with situations in which more than one ball appeared in the scene. In this paper, we extend our previous work to track multiple tennis balls fully automatically. The algorithm presented in this paper requires the set of all-pairs shortest paths in a directed and edge-weighted graph. We also propose an efficient All-Pairs Shortest Path algorithm by exploiting a special topological property of the graph. Comparative experiments show that the proposed data association algorithm performs well both in terms of efficiency and tracking accuracy.

1 Introduction

In automatic video annotation, high-level descriptions rely on low-level features. In the context of a ball game, such as cricket, football, tennis, table tennis or snooker, the trajectory of the ball provides important information for high level annotation. Indeed, reconstructing the ball trajectory is essential for a complete understanding of a ball game. However, tracking a ball in a complex scene can be a difficult task. In the case of tennis ball tracking, the ball's small size, high velocity, abrupt motion change, occlusion, and the presence of multiple balls all pose strong challenges. The scope of this paper is to develop a robust algorithm for tracking tennis balls in broadcast tennis video.

Let us assume we have a ball candidate generation module, where a ball is detected as a candidate with a certain probability along with some clutter-originated false positives. The data association problem, *i.e.* the problem of determining which candidates are ball-originated and which are clutter-originated, is the key problem to solve in tennis ball tracking. In [5], a data association algorithm was proposed under the name of Robust Data Association (RDA), and was used to track a tennis ball. The key idea of RDA is to treat data association as a dynamic model fitting problem. In RDA, a RANSAC [2] paradigm is employed. A sliding window containing several frames is moving over a

sequence. Candidate triplets are randomly drawn from all the candidates in the current interval, and are used to fit dynamic models. The fitted models are evaluated using a cost function. The model is found that is best at explaining the candidates inside the interval. An estimate of the ball position in one frame, *e.g.* the middle frame in the interval, is then given by this model. As the sliding window moves, eventually ball positions in all frames are estimated.

RDA works well under moderate clutter level, and when certain assumptions are satisfied. However, several weaknesses of RDA have been noticed (see [8] for details). Inspired by RDA's model fitting approach to the data association problem, in our previous work [8], we proposed a two-layer data association algorithm (dubbed L^2DA in this paper) to remedy some of the weaknesses of RDA. Although L^2DA provides improved speed and robustness over RDA, like RDA, it is a single-object tracking algorithm, and requires an additional track initiation/termination mechanism. This means L^2DA is not applicable for real world tennis sequences that have complex track initiation/termination scenarios of multiple balls. In this paper, we extend L^2DA to handle multiple objects and to automate the track initiation/termination. This is achieved by using an All-Pairs Shortest Path (APSP) formulation instead of the Single-Pair Shortest Path (SPSP) formulation at the second layer of L^2DA, and by adding a third layer, path level analysis, onto L^2DA. The resulting algorithm is dubbed L^3DA in this paper.

The rest of this paper is organised as follows: Section 2 gives a brief review of L^2DA. Section 3 describes the third layer, path level analysis, of L^3DA. This layer works on the set of all-pairs shortest paths in a graph. In Section 4, we propose an efficient APSP algorithm. Experimental results are presented in Section 5. Finally, conclusions are given in Section 6.

2 The L^2DA Algorithm

In L^2DA, the data association problem is sliced into two layers: candidate level association and tracklet level association. Assume the frames in a sequence are numbered from 1 to K. **At the candidate level**, a sliding window containing $2V+1$ frames is moving over the frames. At time i, the interval I_i centres on frame i and spans frame $i-V$ to frame $i+V$, where $i \in [1+V, K-V]$. Now instead of randomly sampling as in RDA, we exhaustively evaluate for each candidate in frame i whether a small ellipsoid around it in the column-row-time 3D space contains one candidate from frame $i-1$ and one candidate from frame $i+1$. If it does, we call the 3 candidates inside the ellipsoid a "seed triplet", and fit a constant acceleration dynamic model to it. The fitted model is then "improved" by re-fitting another model using candidates in the sliding window that are consistent with it. This process is repeated recursively until convergence, forming what we call a "tracklet": a small segment of a trajectory. Compared to RDA, this "hill-climbing" scheme significantly reduces the algorithm's complexity: as the proportion of true positives drops, the complexity grows approximately linearly.

A tracklet T consists of a parameterised dynamic model M (position and velocity at time i, and the constant acceleration), and a set $\mathscr{S}$ of candidates that support the converged model ("supports" of the model). In other words, $T \triangleq \{M, \mathscr{S}\}$. At time i, there may be multiple tracklets generated. We threshold them based on the number of candidates in their support sets, or their "strengths". Only tracklets that are "strong enough" are

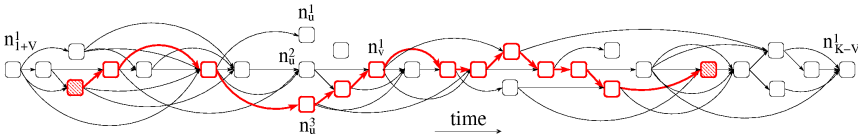


Figure 1: An illustrative example of the topology of $\mathscr{G}$. Each node is a tracklet. Nodes generated in the same sliding window position are aligned vertically. Striped red nodes: the first and last ball-originated nodes. Red nodes and red edges: the shortest path between these two nodes.

retained, and the j^{th} retained tracklet in interval I_i is denoted by $T_i^j = \{M_i^j, \mathscr{S}_i^j\}$.

As the sliding window moves, a sequence of tracklets is generated. These tracklets may have originated from the ball or from clutter. Now we need a data association method **at the tracklet level**. We formulate tracklet level association as a SPSP problem. A directed and edge-weighted graph $\mathscr{G} = \{\mathscr{N}, \mathscr{E}\}$ is constructed, where each node $n_i^j \in \mathscr{N}$ represents the tracklet T_i^j, and the weight $w_{u,v}^{l,m}$ of a directed edge $e_{u,v}^{l,m} \in \mathscr{E}$, which connects n_u^l to n_v^m, is defined according to the "compatibility" of T_u^l and T_v^m, *i.e.* the smaller the $w_{u,v}^{l,m}$, the more likely T_u^l and T_v^m have originated from a same object (see [8] for details). We assume that there is only one ball in the sequence, and that the first and last tracklets (nodes) that have originated from this ball are already known. The ball-originated candidates are then contained in the support sets of the nodes in the shortest path between these two nodes, *i.e.* the path with smallest total edge weight (see Fig. 1).

3 Extending L^{2}DA to L^{3}DA

L^{2}DA assumes there is only one ball to track in a sequence. However, this is not always the case. For example, there may be multiple plays in one sequence, and the second play can start while the ball used for the first play is still in the scene. Moreover, track initiation/termination, which is taken for granted in L^{2}DA, is not a trivial problem, especially when multiple objects are present.

In this section, we extend L^{2}DA to L^{3}DA to deal with multiple objects and to automate the track initiation/termination. This is achieved by using APSP instead of SPSP at the tracklet level, and by introducing one more layer on top of that, namely, path level analysis with a Paths Reduction (PR) algorithm.

For a given pair of nodes n_u^l and n_v^m in $\mathscr{G}$, there may be paths connecting n_u^l to n_v^m, or there may not. Assume the shortest paths between all pairs of nodes that have at least one path connecting them have already been identified. Let $\mathscr{P}$ be the set of such all-pairs shortest paths, and p is the number of paths in $\mathscr{P}$. p is in the order of N^2, where N is the number of nodes in the graph. Now observe that no matter how many balls there are to track, or where each of the ball trajectories starts and terminates in the graph, the paths that correspond to the ball trajectories form a subset of $\mathscr{P}$. The question now is how to reduce the original set of APSP $\mathscr{P}$ to its subset that contains only paths that correspond to the ball trajectories.

We propose a simple Paths Reduction (PR) algorithm to achieve this. The PR algorithm reduces the set of APSP to the Best Set of Compatible Paths (BSCP) $\mathscr{B}$, providing

two assumptions are satisfied: first, the p paths in $\mathscr{P}$ can be ordered according to their "qualities"; and second, a pair-wise "compatibility" of the paths in $\mathscr{P}$ is defined. The PR algorithm is summarised as follows:

- **Initialisation:** $\mathscr{P}$ has p paths, and $\mathscr{B}$ is empty.
- **While** $\mathscr{P}$ is not empty:
  - Remove the best path P^* in $\mathscr{P}$ from $\mathscr{P}$;
  - If P^* is compatible with all paths in $\mathscr{B}$, add P^* to $\mathscr{B}$.

Now we define the relative quality of the paths. Recall that the weight of a path is the sum of the weights of all edges the path goes through. Note that the term "shortest path" used in the previous sections should have been "lightest path". However, we chose to use "shortest path" for the sake of consistency with the terminology used in other papers. We define the strength of a path to be the number of supports in all its nodes, or more precisely, the size of the union of the support sets in all its nodes. Intuitively, a "good" path is one that is both "light" and "strong". However, there is usually a trade-off between the weight and the strength of a path: a stronger path tends to be heavier. Taking this into account, we define the relative quality of two path P_1 and P_2 as follows:

$$P_1 \left\{ \begin{array}{c} > \\ = \\ < \end{array} \right\} P_2 \quad \text{if} \quad (W_1 - W_2) \left\{ \begin{array}{c} < \\ = \\ > \end{array} \right\} \alpha \cdot (S_1 - S_2) \tag{1}$$

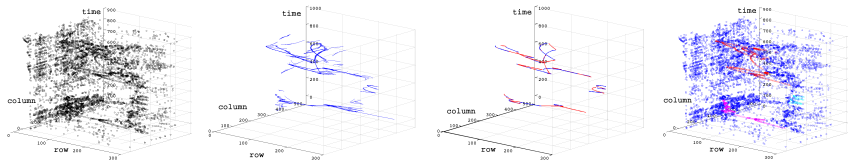
where the relation operators "$>$", "$=$" and "$<$" between P_1 and P_2 stand for "is better than", "has the same quality as", and "is worse than", respectively; W_1 and W_2 are the weights of P_1 and P_2, respectively; S_1 and S_2 are the strengths of P_1 and P_2, respectively; and α is a controllable parameter with the unit of pixel. According to this definition, if a path P_1 is "much stronger" but "slightly heavier" than a path P_2, then P_1 is said to have a better quality than P_2. Note that this definition does not assume any relationship between W_1 and W_2, or relationship between S_1 and S_2.

It easily follows that the set $\mathscr{P}$ equipped with an operator "$\geq$" satisfies the following three statements:

1. **Transitivity:** if $P_1 \geq P_2$ and $P_2 \geq P_3$ then $P_1 \geq P_3$;
2. **Antisymmetry:** if $P_1 \geq P_2$ and $P_2 \geq P_1$ then $P_1 = P_2$;
3. **Totality:** $P_1 \geq P_2$ or $P_2 \geq P_1$.

According to order theory [1], $\mathscr{P}$ associated with operator "$\geq$" is a totally ordered set. The first assumption for the PR algorithm to work is satisfied.

The second assumption, the existence of pair-wise compatibility of the paths, is straightforward. Two paths are said to be compatible if and only if they do not share any common **support**. It should be noted, however, two paths that do not share any common **node** are not necessarily compatible, because different nodes can have common supports.



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

Figure 2: (a): ball candidates in an example sequence. Each black circle is a candidate. (b): generated tracklets. (c): results of applying APSP and the PR algorithm: 3 paths in $\mathscr{B}_{th}$. Adjacent nodes in each path are plotted alternatively in blue and red. (d): recovered class labels as given by $\mathscr{B}_{th}$.

Figure 3: Ball trajectories (after interpolation and key event detection) superimposed on mosaic images. From left to right: the first, second and third play in time order.

Now with SPSP replaced by APSP at the tracklet level, and with the PR algorithm at the path level, we have extended L^2DA to L^3DA. We apply L^3DA to an example sequence (see Fig. 2). Semantically, the ball-originated candidates in this sequence belong to three plays. In time order (from bottom to top in the figures), the first play (magenta circles in Fig. 2 (d)) is a bad serve, where the ball lands outside the service box; the second "play" (cyan circles in Fig. 2 (d)) is a player bouncing the ball on the ground preparing for the next serve; and the third play (red circles in Fig. 2 (d)) is a relatively long one with several exchanges. The objective of data association is to identify the number of plays in this sequence, and to recover the class label of each candidate: clutter, first play, second play, or third play. In other words, the objective is to recover the colour information in Fig. 2 (d), assuming it is lost (see Fig. 2 (a)).

First, we "grow" tracklets from seed triplets (see Fig. 2 (b)), as in L^2DA. By looking for all-pairs shortest paths, a set $\mathscr{P}$ with $p = 87961$ paths is obtained. The PR algorithm is then applied, which gives a BSCP $\mathscr{B}$ containing 11 paths. In descending order, the numbers of supports (strengths) of the paths in $\mathscr{B}$ are: 411, 247, 62, 23, 20, 17, 17, 16, 15, 10, 9. It is a reasonable assumption that a path corresponding to a ball trajectory has more supports than a path corresponding to the motion of a non-ball object, *e.g.* a wristband worn by a player (which can be detected as ball candidates and can form smooth trajectory as the player strikes the ball). We set a threshold S_{th} and keep only the

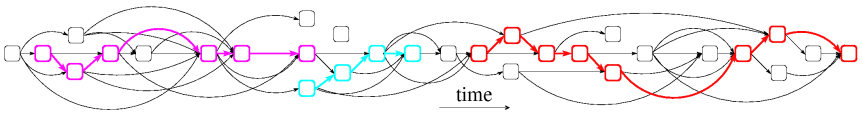


Figure 4: A possible arrangement of the paths in $\mathscr{B}_{th}$. Magenta, cyan, and red paths correspond to the first, second and third play in the sequence, respectively.

paths that have more supports than S_{th}. This results in a thresholded BSCP $\mathscr{B}_{th}$ with 3 paths (see Fig. 2 (c)), where each path corresponds to a play in the sequence.

In tennis ball tracking, the points at which the ball changes its motion abruptly correspond to key events such as hit and bounce, and provide important information for high level annotation. We detect these key events by looking for motion discontinuities in the trajectories. In Fig. 3, the 3 ball trajectories after interpolation and event detection are superimposed on mosaic images.

A suggestion of how the 3 paths might be arranged in the graph $\mathscr{G}$ is shown in Fig. 4. Note that there are 672 tracklets in this sequence. Far fewer nodes are plotted in Fig. 4 for ease of visualisation. Note also that two paths that are temporally overlapping are not necessarily incompatible. In fact, the first and second plays in the example sequence do overlap in time: the first play spans frame 16 to frame 260, and the second spans frame 254 to frame 321.

4 An Efficient APSP Algorithm

In L^{3}DA, at the tracklet level, we need to solve an APSP problem for a graph $\mathscr{G}$ with N nodes. In some sequences, N can be in the order of 10^3. An efficient APSP algorithm is desirable. Several APSP algorithms have been reported in the literature. The Floyd-Warshall algorithm solves APSP in $O(N^3)$ time [3]. Johnson's algorithm has a complexity of $O(N^2 \log N + NE)$, where E is the number of edges in the graph [4]. Neither the Floyd-Warshall algorithm nor Johnson's algorithm makes any assumption about the topology of the graph. Because of the way our graph is constructed, it has a special topological property: its set of nodes $\mathscr{N}$ can be partitioned into subsets $\mathscr{N}_{1+V}, \mathscr{N}_{2+V}, ..., \mathscr{N}_{K-V-1}, \mathscr{N}_{K-V}$, where $\mathscr{N}_i$ is the set of nodes generated in interval I_i, such that edges exist from nodes in subset $\mathscr{N}_u$ to nodes in subset $\mathscr{N}_v$ only if $u < v$ (see [8] for details). Using this property, we derive an $O(N^2)$ APSP algorithm as follows.

The proposed APSP algorithm uses the concept of dynamic programming. Suppose we are in the middle of the tracklet generation process. The sliding window now centres on frame $i-1$, and tracklets in interval I_{i-1} have been generated. Let $\mathscr{G}^{(i-1)} = \{\mathscr{N}^{(i-1)}, \mathscr{E}^{(i-1)}\}$ be the graph constructed so far, where $\mathscr{N}^{(i-1)} = \{\mathscr{N}_{1+V}, \mathscr{N}_{2+V}, ..., \mathscr{N}_{i-1}\}$; $\mathscr{E}^{(i-1)}$ is the set of edges that go into all nodes in $\mathscr{N}^{(i-1)}$. Clearly, $\mathscr{G}^{(i-1)}$ is a sub-graph of the complete graph $\mathscr{G}$. Assume the APSP problem in graph $\mathscr{G}^{(i-1)}$ has been solved. That is, in each node $n_v^m \in \mathscr{G}^{(i-1)}$, a table is maintained, where each entry corresponds to a node in the sub-graph $\mathscr{G}^{(v-1)}$. The entry corresponding to node $n_u^l \in \mathscr{G}^{(v-1)}$ keeps two pieces of information about the shortest path from n_u^l to n_v^m in $\mathscr{G}^{(i-1)}$. The first one is the last node before n_v^m in the shortest path, and the second one is the total weight of the shortest path. With these two pieces of information for each node $n_u^l \in \mathscr{G}^{(v-1)}$ in each node $n_v^m \in \mathscr{G}^{(i-1)}$, the shortest path between any pair of nodes in $\mathscr{G}^{(i-1)}$ can be identified by back tracing.

Next, we show how to solve the APSP problem in $\mathscr{G}^{(i)}$ using the solution of the APSP problem in $\mathscr{G}^{(i-1)}$. Now the sliding window moves one frame forward, and the interval I_i centres on frame i. Assume several tracklets are generated in I_i, forming the set of nodes $\mathscr{N}_i$. Now we need to construct for each node $n_i^j \in \mathscr{N}_i$ a table of APSP knowledge, where each entry contains information about the shortest path in $\mathscr{G}^{(i)}$ from a node in $\mathscr{G}^{(i-1)}$ to n_i^j.

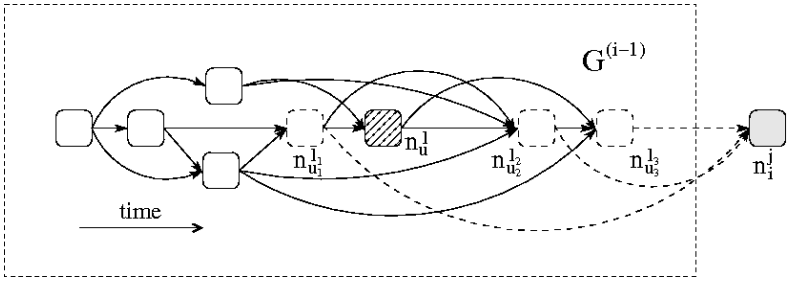


Figure 5: Constructing the table of APSP knowledge for a node $n_i^j \in \mathscr{N}_i$.

In Fig. 5, the sub-graph inside the big rectangle represents $\mathscr{G}^{(i-1)}$, and a new node $n_i^j \in \mathscr{N}_i$ is plotted as a shaded node. Assume s nodes in $\mathscr{G}^{(i-1)}$ are connected to n_i^j with edges. These s nodes are denoted by $n_{u_1}^{l_1}, n_{u_2}^{l_2}, ..., n_{u_s}^{l_s}$, and are plotted as dashed nodes in Fig. 5. Edges that connect these nodes to n_i^j are denoted by $e_{u_1,i}^{l_1,j}, e_{u_2,i}^{l_2,j}, ..., e_{u_s,i}^{l_s,j}$, and are plotted as dashed edges. Obviously, the number of entries in the table of APSP knowledge in n_i^j is equal to the number of nodes in $\mathscr{G}^{(i-1)}$. Without loss of generality, let us consider one entry in the table, which keeps information about the shortest path in $\mathscr{G}^{(i)}$ from a node $n_u^l \in \mathscr{G}^{(i-1)}$ to n_i^j. In Fig. 5, n_u^l is plotted as a striped node. Now observe that the shortest path in $\mathscr{G}^{(i)}$ from n_u^l to n_i^j must go through one of the nodes in $n_{u_1}^{l_1}, n_{u_2}^{l_2}, ..., n_{u_s}^{l_s}$ and the corresponding edge in $e_{u_1,i}^{l_1,j}, e_{u_2,i}^{l_2,j}, ..., e_{u_s,i}^{l_s,j}$. Since APSP has been solved in $\mathscr{G}^{(i-1)}$, the information about the shortest path in $\mathscr{G}^{(i-1)}$ from n_u^l to $n_{u_r}^{l_r}$ is kept in the table in $n_{u_r}^{l_r}$, where $r = 1, 2, ..., s$. Let $W^{(i-1)}(n_u^l, n_{u_r}^{l_r})$ be the total weight of the shortest path in $\mathscr{G}^{(i-1)}$ from n_u^l to $n_{u_r}^{l_r}$, as kept in the table in $n_{u_r}^{l_r}$. Specially, if the table in $n_{u_r}^{l_r}$ does not contain an entry for n_u^l, it means $u_r \leq u$, and we define for this case $W^{(i-1)}(n_u^l, n_{u_r}^{l_r}) = \infty$. The total weight of the shortest path in $\mathscr{G}^{(i)}$ from n_u^l to n_i^j is then:

$$W^{(i)}(n_u^l, n_i^j) = \min[W^{(i-1)}(n_u^l, n_{u_r}^{l_r}) + w_{u_r,i}^{l_r,j}], \forall r \in [1, s] \tag{2}$$

where $w_{u_r,i}^{l_r,j}$ is the weight of edge $e_{u_r,i}^{l_r,j}$. The last node before n_i^j in the shortest path in $\mathscr{G}^{(i)}$ from n_u^l to n_i^j is $n_{u^*}^{l^*}$, where

$$\{u^*, l^*\} = \arg \min_{\{u_r, l_r\}} [W^{(i-1)}(n_u^l, n_{u_r}^{l_r}) + w_{u_r,i}^{l_r,j}], \forall r \in [1, s] \tag{3}$$

The two pieces of information for one entry in the table in node n_i^j are thus obtained: $W^{(i)}(n_u^l, n_i^j)$ and $n_{u^*}^{l^*}$ are put into the entry for n_u^l. This process is applied to each node in $\mathscr{G}^{(i-1)}$, whereupon the complete table in n_i^j is constructed. Using the special topological property of the graph $\mathscr{G}$ discussed at the beginning of this section, the shortest path in $\mathscr{G}^{(i-1)}$ between any pair of nodes in $\mathscr{G}^{(i-1)}$ is also the shortest path in $\mathscr{G}^{(i)}$ between the same pair of nodes. When the new node n_i^j and the associated edges $e_{u_1,i}^{l_1,j}, e_{u_2,i}^{l_2,j}, ..., e_{u_s,i}^{l_s,j}$ are added to $\mathscr{G}^{(i-1)}$, the tables in the nodes in $\mathscr{G}^{(i-1)}$ remain the same. This means that, simply by applying the above process as new nodes (and associated edges) are received, when the complete graph $\mathscr{G} = \mathscr{G}^{(K-V)}$ is constructed, the APSP problem in it is solved.

SVM boundary	-4	-3	-2	-1	0	1
r_d	0.917	0.916	0.908	0.874	0.822	0.531
$\bar{N}$	12.2	9.0	5.1	0.9	0.1	0

Table 1: Detection rate and clutter level with various SVM boundaries.

The shortest path between any pair of nodes in $\mathscr{G}$ can be easily identified by back tracing. The proposed APSP algorithm is summarised as follows:

- **Assume:** the APSP problem in $\mathscr{G}^{(i-1)}$ has been solved.
- **For** each node $n_i^j \in \mathscr{N}_i$:
  - **For** each node $n_u^l \in \mathscr{G}^{(i-1)}$:
    - add an entry labelled n_u^l to the table of APSP knowledge in n_i^j;
    - put $W^{(i)}(n_u^l, n_i^j)$ and $n_{u^*}^{l^*}$ given by (2) and (3) into this entry.

Let h_i be the number of nodes in $\mathscr{N}_i$. The number of nodes in sub-graph $\mathscr{G}^{(i-1)}$ is then $\sum_{k=1+V}^{i-1} h_k$. To solve the APSP problem in $\mathscr{G}^{(i)}$, we need to construct a table of APSP knowledge for each node in $\mathscr{N}_i$. The number of operations of this process is in the order of $h_i \sum_{k=1+V}^{i-1} h_k$. The number of operations of the proposed APSP algorithm is then in the order of $\sum_{i=2+V}^{K-V}(h_i \sum_{k=1+V}^{i-1} h_k)$. Simple manipulation shows that the complexity of the proposed APSP algorithm is $O(N^2)$, where $N = \sum_{i=1+V}^{K-V} h_i$ is the number of nodes in $\mathscr{G}$.

5 Experiments

We used 60 sequences from the 2006 Australia Open tournament Men's final game for our experiments. The number of plays in each sequence ranges from 2 to 4. In total the 60 sequences are approximately 16 minutes long, and contain 50,662 frames.

We used frame differencing to extract foreground moving objects. A Support Vector Machine (SVM) was trained and used to classify the foreground blobs into ball candidates and non-candidates. Features used in the SVM are the shape, colour and position of each blob. By moving the decision boundary of the SVM, a trade-off can be made between the ball detection rate r_d and the average number of false candidates $\bar{N}$ in each frame. Table 1 shows 6 SVM boundaries and the corresponding r_d and $\bar{N}$. Using these 6 configurations, we can evaluate a tracker's performance under various detection rate and clutter level.

RDA and another two tennis ball tracking algorithms from our previous work [6, 7], one based on particle filtering, and the other based on the Viterbi algorithm, were also implemented for comparison. For these three trackers, one instance of the tracker was used to track each play in each sequence, and track initiation/termination of each play was manually dealt with. In RDA, the number of trials, N_t, is chosen so that the probability of finding a set that consists entirely of true positives is greater than a threshold γ. In our experiments, γ was set to 0.99.

SVM boundary		-4	-3	-2	-1	0	1
prop. of LOT Frames	particle	8.64%	9.61%	6.92%	6.63%	8.29%	19.16%
	Viterbi	**4.14%**	3.88%	**3.41%**	3.23%	3.64%	4.12%
	RDA	17.06%	15.73%	12.43%	9.18%	6.99%	7.27%
	L^3DA	4.40%	**3.68%**	3.57%	**2.81%**	**2.41%**	**2.73%**

Table 2: Proportion of loss-of-track (LOT) frames.

SVM boundary		-4	-3	-2	-1	0	1
processing speed (frames per sec)	particle	21.0	23.2	25.1	26.6	28.8	30.7
	Viterbi	31.3	36.7	40.4	42.2	45.9	47.3
	RDA	0.9	1.7	23.5	**233.0**	**374.8**	**399.1**
	L^3DA	**46.3**	**59.4**	**72.8**	93.6	116.2	142.4

Table 3: Processing speed.

To evaluate the performance of the trackers, ground truth of the tennis ball positions in all frames was manually marked. Tracking results were then compared against the ground truth. Tracking error is defined as the Euclidean distance between the ground truth and the tracked (detected or interpolated) ball position. A loss-of-track (LOT) frame is defined as a frame where the tracking error is greater than 6 pixels. Table. 2 shows the proportion of LOT frames of each tracker with each SVM boundary. In brief, L^3DA and the Viterbi-based tracker outperform the other two trackers. When looking more carefully at Table. 2, we can see the four algorithms have different failure modes.

When r_d and $\bar{N}$ are both low, the particle-base algorithm performs poorly. This is because the ball changes its motion drastically after being hit by a player. Consequently, the next detected ball-originated candidate can be very far from its predicted position. This is especially the case when r_d is low. As a result, the particle-based tracker can be "trapped" by false candidates that have originated from the player, and cannot recover until the ball is close to the player again. On the other hand, L^3DA, being a non-iterative algorithm, is much more robust against sudden change of motion direction.

RDA performs poorly when r_d and $\bar{N}$ are high. This is because in RDA, or more generally in RANSAC, we make the implicit assumption that a model given by an uncontaminated sample set is always "better" than that given by a contaminated sample set. However, in a tennis sequence, especially when multiple balls are present, the ball being tracked is not the only smoothly moving object. Candidates that have originated from other balls, or even from part of a player, *e.g.* a wrist band, can form smooth trajectories. As a result, a model given by candidates that have originated from the ball being tracked can "lose" in the competition with a model given by candidates that have originated from other objects. This problem is tackled in L^3DA by enforcing motion consistency with the shortest path formulation.

The Viterbi-based algorithm gives similar performance to that of L^3DA. However, L^3DA has the advantage of being fully-automatic, while the Viterbi-based algorithm requires an additional track initiation/termination mechanism.

In Table 3, the speed of the four algorithms is compared. L^3DA shares the top position with the LDA. The fact that L^3DA always starts model fitting from a seed triplet — three candidates that have high probability of containing only true positives — allows it to eliminate false candidates very quickly. The proposed APSP algorithm also helps improve the efficiency of L^3DA. It should be noted that as the SVM boundary increases, RDA has the fastest growing processing speed. This is because the time complexity of RDA is determined directly by N_t, which drops rapidly as the proportion of true positives increases.

6 Conclusions

In this paper, we have extended our previous work L^2DA, a semi-automatic single-object tracking algorithm, to L^3DA, a fully automatic multiple-object tracking algorithm. This was achieve by using APSP instead of SPSP at the tracklet level, and by adding one more layer, path level analysis, on top of L^2DA. In this paper, we have also proposed an efficient APSP algorithm by exploiting a special topological property of the graph. The proposed L^3DA algorithm was used to track tennis balls in broadcast tennis video. Comparative experiments show that it performs well both in terms of efficiency and tracking accuracy.

Acknowledgements

This work has been supported by the EU IST-507752 MUSCLE Network of Excellence.

References

[1] B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, 2002.

[2] 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 Association for Computing Machinery*, 24(6):381–395, 1981.

[3] R. W. Floyd. Algorithm 97: Shortest path. *Communications of the ACM*, 5(6):345, 1962.

[4] D. B. Johnson. Efficient algorithms for shortest paths in sparse networks. *Journal of the ACM*, 24(1):1–13, 1977.

[5] V. Lepetit, A. Shahrokni, and P. Fua. Robust data association for online applications. In *IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 281–288, 2003.

[6] F. Yan, W. Christmas, and J. Kittler. A tennis ball tracking algorithm for automatic annotation of tennis match. In *British Machine Vision Conference*, volume 2, pages 619–628, 2005.

[7] F. Yan, W. Christmas, and J. Kittler. A maximum a posteriori probability viterbi data association algorithm for ball tracking in sports video. In *IEEE International Conference on Pattern Recognition*, 2006.

[8] F. Yan, A. Kostin, W. Christmas, and J. Kittler. A novel data association algorithm for object tracking in clutter with application to tennis video analysis. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2006.

Shape from Texture: Fast Estimation of Planar Surface Orientation via Fourier Analysis

Fabio Galasso and Joan Lasenby
Department of Engineering, University of Cambridge
Cambridge, CB2 1PZ, UK
fg257@cam.ac.uk, jl221@cam.ac.uk

Abstract

Shape from texture has received much attention in the past few decades. We propose a computationally efficient method to extract 3D planar surface orientation from the spectral variations of a visual texture. Under the assumption of homogeneity, the texture is represented by the novel method of identifying ridges of its Fourier transform. Local spatial frequencies are then computed using a minimal set of selected Gabor filters. Under perspective projection, frequencies are backprojected and orientation is computed so as to minimize the variance of the frequencies' backprojections. A comparative study with two existing methods, and experimentation on simulated and real texture images is given.

1 Introduction

Shape from Texture was first introduced by Gibson 50 years ago. In [7] he suggests that texture can provide an important shape cue. However for a machine the solution to this problem is ill-posed. Shape from texture is generally about measuring the texture distortion in an image, and then reconstructing the surface 3D coordinates in the scene ([6], [8], [9], [11]). The model for the texture can be either deterministic or stochastic. The second allows a wider variety of textures ([9], [11], [13]) and implies local spectral measurements, usually with the Fourier transform ([11]), or more recently, wavelets ([8], [3]).

An initial assumption about the texture is always necessary, and few of the existing papers are applicable to real surfaces because of restrictive assumptions. [10] deals with texels, which are seldom found in nature, while [14] assumes isotropy, rarely the case. Homogeneity is more frequently used ([9], [6], [3]), and is the one we choose here. For deterministic textures it can be seen as periodicity, for stochastic textures it can be formalized as stationarity under translation ([11]). Under this condition we assume that all texture variations are produced only by projective geometry.

We assume here a perspective or pin-hole camera model, as in [4] and [12], because perspective effects (e.g. shrinking) are usually found in images of slanted planes. We do not consider the weak perspective case as this preserves homogeneity and therefore gives no information on plane orientation ([5] and references within).

The present work takes its motivation from [12]. The texture is analyzed using Gabor filters to produce distortion information based on *local spatial frequency* (LSF). Unlike

[12], we do not just rely on a dominant LSF, but we consider groups of LSFs. This extends [12] to exploit the multi-scale nature of textures. To our knowledge the algorithm presented here is the first to consider the multi-scale nature of texture to the extent of exploiting all main LSFs, most of the related work uses only two preferred directions in the spectral domain (e.g. [13]).

Section 2 explains in detail how the texture is analyzed to produce distortion information, and justifies the chosen method. Section 3 presents the projective geometry. Section 4 shows how we can recover surface 3D coordinates from the measured texture distortion. Finally, section 5 presents results, comparing them with those in [8].

2 Texture Description

Here we describe how to set 2D Gabor functions and their first derivatives from the information on texture supplied by the Fourier transform. The former provide local analysis to compute instantaneous frequencies, which are used to measure distortion and reconstruct the 3D coordinates of the texture surface.

2.1 Estimating the Instantaneous Frequencies

The analysis of an image $I(\mathbf{x})$ is usually done using a band-pass filter $h(\mathbf{x},\mathbf{u})$, a function of a point $\mathbf{x} = (x,y)$ and of a central frequency $\mathbf{u} = (u,v)$, which is convolved with the image to provide the local spectrum. As in [12] we choose 2D Gabor functions:

$$h(\mathbf{x},\mathbf{u}) = g(\mathbf{x})e^{2\pi j\mathbf{x}\cdot\mathbf{u}} \quad \text{where} \quad g(\mathbf{x}) = \frac{1}{2\pi\gamma^2}e^{\frac{-(\mathbf{x}\cdot\mathbf{x})}{2\gamma^2}} \tag{1}$$

with j the unit imaginary and $g(\mathbf{x})$ a 2D Gaussian function with variance γ^2.

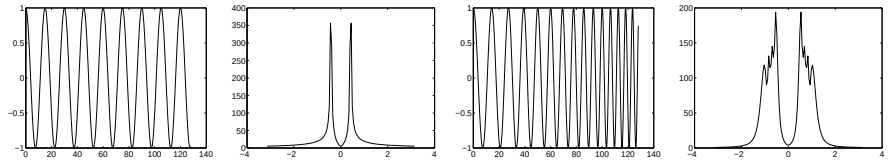
For a 2D cosine $f(\mathbf{x}) = \cos(2\pi\Omega(\mathbf{x}))$ the instantaneous frequency is given by

$$\tilde{\mathbf{u}}(\mathbf{x}) = (\tilde{u}(\mathbf{x}),\tilde{v}(\mathbf{x})) = \left(\frac{\partial\Omega}{\partial x},\frac{\partial\Omega}{\partial y}\right). \tag{2}$$

Our goal is to measure $\tilde{\mathbf{u}}(\mathbf{x})$. [1] shows that this can be done by considering a Gabor function $h(\mathbf{x},\mathbf{u})$, and its two first order derivatives, $h_x(\mathbf{x},\mathbf{u})$ and $h_y(\mathbf{x},\mathbf{u})$:

$$\begin{aligned} |\tilde{u}(\mathbf{x})| &= \frac{|h_x(\mathbf{x},\mathbf{u}) * I(\mathbf{x})|}{2\pi|h(\mathbf{x},\mathbf{u}) * I(\mathbf{x})|} \\ |\tilde{v}(\mathbf{x})| &= \frac{|h_y(\mathbf{x},\mathbf{u}) * I(\mathbf{x})|}{2\pi|h(\mathbf{x},\mathbf{u}) * I(\mathbf{x})|}. \end{aligned} \tag{3}$$

This estimate can be assumed to be correct if the frequency we are measuring is in the pass-band of the filter. This method implies that we have to choose the central frequencies $\mathbf{u}$ of the Gabor functions, and the spatial constants γ, in order to set the centre and width of the filters. The filters have constant *fractional bandwidth* (bandwidth divided by its centre frequency). This allows us to measure higher frequencies more locally than lower frequencies and is computationally less expensive. Moreover, as all filters so derived are geometrically similar it is simpler to compare their outputs.

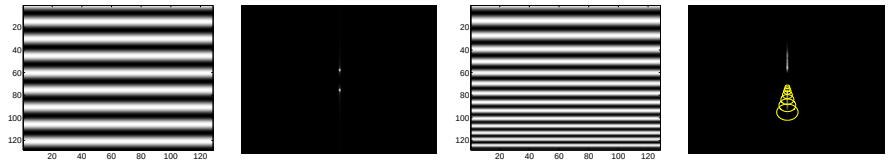


(a) 1D cosine at frequency $\tilde{u} \approx 0.42$ rad/s

(b) Amplitude of the spectrum of the cosine in 1(a)

(c) 1D cosine with frequency varying from $\tilde{u} \approx 0.42$ rad/s to $\tilde{u} \approx 1.27$ rad/s

(d) Amplitude of the spectrum of the cosine in 1(c)

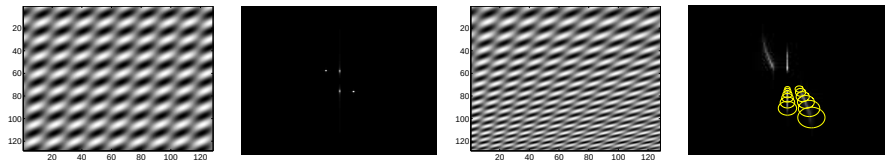


(e) 2D cosine at frequency $|\tilde{\mathbf{u}}| \approx 0.42$ rad/s

(f) Amplitude of the spectrum of the cosine in 1(e)

(g) 2D cosine with frequency continuously varying from $|\tilde{\mathbf{u}}| \approx 0.42$ rad/s to $|\tilde{\mathbf{u}}| \approx 1.27$ rad/s

(h) Amplitude of the spectrum of 1(g) and the chosen set of Gabor filters (circles) on it



(i) Two 2D cosines superposed, at frequencies $|\tilde{\mathbf{u}}_1| \approx 0.42$ rad/s and $|\tilde{\mathbf{u}}_2| \approx 0.63$ rad/s, 45° degrees apart

(j) Amplitude of the spectrum of the image in 1(i)

(k) Result of slanting the image in 1(i) by 38°

(l) Amplitude of the spectrum in 1(k) and the chosen set of Gabor filters (circles) on it

Figure 1: Setting the Gabor filters' parameters

We choose to set the Gabor functions using the information from the Fourier transform of the texture. Unlike Super and Bovik ([12]), who sample the whole 2D frequency plane, we make a selection of Gabor filters using ridges in the Fourier transform of the image. In our algorithm every ridge determines a set of Gabor filters that covers the corresponding values of frequencies. Every ridge therefore determines different instantaneous frequencies and thus different distortion measures.

2.2 Setting the Gabor Filter Parameters

Let us consider a 1D cosine (figure 1(a)). The signal has length of 128 samples and frequency $\tilde{u} \approx 0.42$ rad/s (where π rad/s is by convention the biggest admissible frequency). Figure 1(b) represents its spectrum amplitude, two symmetric spikes at the corresponding frequencies ($\approx \pm 0.42$ rad/s). A chirp is shown in figure 1(c), i.e. a cosine with frequency varying from $\tilde{u} \approx 0.42$ rad/s to $\tilde{u} \approx 1.27$ rad/s. Figure 1(d) illustrates its spectrum, where significant non-zero values span that range.

Analogously we show a 2D image generated by a 2D cosine with frequency $|\tilde{\mathbf{u}}| \approx 0.42$ rad/s (figure 1(e)) and its spectrum (figure 1(f)), given by symmetric spikes on the fre-

quency value. And we compare them to figures 1(g) and 1(h), the image of a 2D cosine with frequency ranging from $|\tilde{\mathbf{u}}| \approx 0.42$ rad/s to $|\tilde{\mathbf{u}}| \approx 1.27$ rad/s and its spectrum (circles are fully explained later). In the latter, significant non-zero values form a ridge corresponding to that range. Figures 1(g) and 1(h) were actually generated by slanting (see section 3) image 1(e) through 38°. The ridges of the amplitude of the Fourier transform of the image represent the 2D frequencies contained in the texture.

The algorithm we propose analyzes the spectrum of the texture to determine its ridges, and then uses this information to define the sets of Gabor functions used. Figure 1(h) shows the chosen set of central frequencies $\mathbf{u}$ (the centres of the circles) and the set of spatial constants γ (their radii); half of the spectrum is considered because of its redundancy. There is significant overlapping (50%) to produce a robust LSF estimation. However, unlike in [12], where 63 central frequencies and spatial constants sample the whole 2D frequency plane, here the number used varies with the image. 7 $\mathbf{u}$'s and γ's are used in figure 1(h). This implies a significant reduction of the computational expense: in [12] 63 $\mathbf{u}$'s and γ's correspond to 378 convolutions (the Gabor filter and its first order derivatives and an equivalent number of post-smoothing filters); our algorithm in this case uses 7 $\mathbf{u}$'s and γ's, meaning 42 convolutions, therefore a computational saving of about 89%.

We now consider the case of multiple frequencies. Figure 1(i) shows the cosine from the previous example ($|\tilde{\mathbf{u}}_1| \approx 0.42$ rad/s) on which we have superposed another cosine, with frequency $|\tilde{\mathbf{u}}_2| \approx 0.63$ rad/s, separated by 45° degrees from the first in the frequency plane. The amplitude of the spectrum of the image (figure 1(j)) shows four peaks, corresponding to the values of the two frequencies of the cosines. In this case we can associate two instantaneous frequencies to each point, which in fact coexist at every pixel. Figure 1(k) shows the result of applying the same slant as in figure 1(g): each cosine has now a continuously-varying frequency. Moreover the two LSFs change independently from each other. In fact the first cosine acquires the same continuously-varying frequency as in the previous section, and the second equivalently acquires a range of 2D frequencies varying in the direction of the slant. This is what the amplitude of the spectrum in figure 1(l) shows. In it we can observe two ridges, each of them associated with the original cosines, the spread indicating a variation or distortion due to the slant.

Our algorithm detects the two ridges and sets two groups of Gabor filters. In each group a series of values for the central frequencies, $\mathbf{u}$'s, and the spatial constants, γ's, are defined, so as to determine the filters to cover the respective ridge area (figure 1(l)). Every set of filters is processed as in the previous example, i.e. as if the texture contained only one corresponding LSF. Thus each set of filters reconstructs an instantaneous frequency for each pixel. These are used to measure the deformation of the texture due to the slanting, are processed independently and finally the results are combined (details are in section 4). In this sense we exploit the multi-scale nature of the texture, because all different-scale frequencies are considered in the final result.

3 Projection of Texture

Here we describe the viewing geometry and a projection model, to provide a relationship between the surface and the image plane as a function of the orientation. We then present a surface texture model.

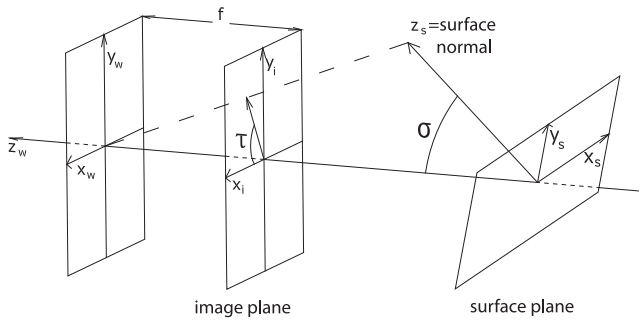


Figure 2: Viewing geometry and projection model

3.1 Viewing Geometry and Projection Model

We adopt the viewing geometry and projection model of [12]. They assume a pin-hole camera model and their coordinate systems are given in figure 2. In it the origin of the world coordinate system $\mathbf{x}_w = (x_w, y_w, z_w)$ coincides with the focal point and the optical axis coincides with the $-z_w$-direction. The image plane coordinate system $\mathbf{x}_i = (x_i, y_i)$ is placed at $z = f < 0$, with $|f|$ being the focal length, such that $x_i = x_w$ and $y_i = y_w$. The orientation of the surface is described using the slant-tilt system: the slant σ is the angle between the surface normal and the optical axis, with values ranging from 0° to 90°; the tilt τ is the angle between the x_i-axis and the projection on the image plane of the surface normal, with values between -180° and 180°. The surface is described by the coordinate system $\mathbf{x}_s = (x_s, y_s, z_s)$: the x_s-axis is aligned with the perceived tilt direction, the z_s-axis is aligned with the surface normal, y_s forms a right handed orthogonal coordinate system and the origin of $\mathbf{x}_s$ is on the intersection of the surface with the z_w-axis, at $z_w = z_0 < 0$.

[12], to which we refer for details of the derivation, obtains the equations for transforming 2D surface to 2D image coordinates, and vice versa, under perspective projection. Most importantly, they derive the relationship between the instantaneous frequencies on the image plane $\mathbf{u}_i = (u_i, v_i)$ and those on the surface plane $\mathbf{u}_s = (u_s, v_s)$:

$$\mathbf{u}_s = J^t(\mathbf{x}_i, \mathbf{x}_s) \cdot \mathbf{u}_i. \tag{4}$$

J^t, the transpose of the Jacobian determinant of the coordinate transformation, is

$$J^t(\mathbf{x}_i, \mathbf{x}_s) = \frac{\sin\sigma}{z_w} \begin{bmatrix} x_i & y_i \\ 0 & 0 \end{bmatrix} + \frac{f}{z_w} \begin{bmatrix} \cos\sigma\cos\tau & \cos\sigma\sin\tau \\ -\sin\tau & \cos\tau \end{bmatrix} \tag{5}$$

$$\text{with} \quad z_w = z_0 - x_s \sin\sigma = \frac{f z_0 \cos\sigma}{\sin\sigma(x_i\cos\tau + y_i\sin\tau) + f\cos\sigma}. \tag{6}$$

We use the above to backproject a LSF computed on the image plane to the surface plane.

3.2 Surface Texture Model

We model textures as due to variations of surface reflectance, the proportion of incident light reflected. We assume that the surfaces have a Lambertian reflection, and that the texture is therefore 'painted' on them, without roughness or self-occlusion.

Surface reflectance, $t_s(\mathbf{x}_s)$, and image reflectance, $t_i(\mathbf{x}_i)$, are related by the following:

$$t_i(\mathbf{x}_i) = k(\mathbf{x}_i) \cdot t_s[\mathbf{x}_s(\mathbf{x}_i)], \tag{7}$$

where $\mathbf{x}_s(\mathbf{x}_i)$ represents the perspective backprojection, while $k(\mathbf{x}_i)$ is a multiplicative shading term. [4] shows how to estimate and remove k. However, if the scale of variation of t_s is small compared to the scale of variation of the shading term, then the latter can be assumed to be constant in any small neighborhood. Moreover, our method automatically normalizes for slow variations in illumination, shading and surface texture contrast, because it uses frequencies rather than amplitudes. Also no assumption is made about the textural nature of $t_s(\mathbf{x}_s)$, thus it might apply to various patterns, e.g. lines, blobs, etc.

4 Computing Surface Orientation

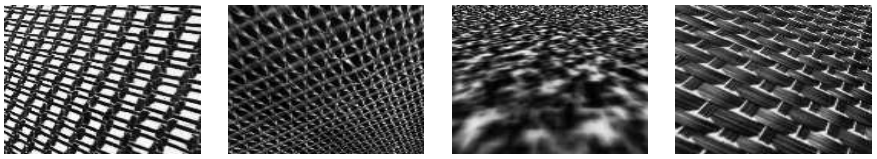
We explain here how our algorithm processes the image texture to produce the orientation of the surface texture.

As discussed in the introduction, we assume homogeneity, in the specific form that the relevant LSFs of the textured surface are constant in the surface region under analysis. Our assumption includes as a corollary that the variance of each LSF on the surface plane is zero. The theoretical zero value means a minimum in the case of real data, and this assumption is used to compute the surface orientation, i.e. the slant σ and tilt τ.

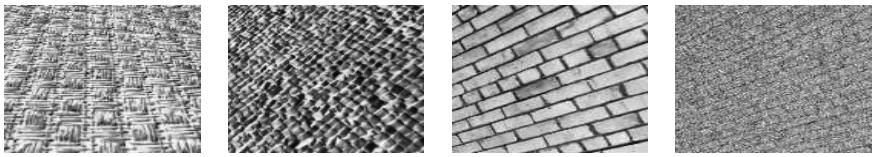
The structure of the proposed algorithm is therefore:

- The spectrum amplitude of the image texture is analyzed and ridges are detected.
- Each ridge determines a set of Gabor functions and their first derivatives, so that the filters cover the frequencies pertaining to the particular ridge.
- For each set of filters the following steps are repeated:
  - the image is convolved with the Gabor filters and their derivatives, and the outputs are smoothed with a Gaussian to reduce noise;
  - the Gabor filter with largest amplitude output is selected at each point;
  - the (signed) instantaneous frequencies are computed at each point (eq. 3);
  - a 2D search over the plane σ-τ is implemented: for each couple (σ,τ) the instantaneous frequency is backprojected using equation 4, and the variance $V_{\sigma,\tau}$ is computed;
  - the values of σ and τ corresponding to the minimum variance are chosen, and the variance is also returned.
- The algorithm chooses the best couple (σ,τ) as that giving the lowest variance.

The minimum variance ($V_{\sigma,\tau}$) method requires the estimated instantaneous frequencies to pertain to the same slanted and tilted LSF in every group. This is not assured if we use a grid of Gabor filters and choose the largest amplitude output, as in [12]. In this case, maximum outputs might then correspond to different groups of LSFs for different pixels in textures with more than one dominant frequency, which invalidates the orientation estimation. Our algorithm allows us to estimate instantaneous frequencies pertaining to distinct groups because it uses separated sets of filters. This improves its robustness.



(a) D20 (b) D52 (c) D57 (d) D65



(e) D82 (f) D84 (g) D95 (h) D95 with white Gaussian noise (SNR=−5dB)

Figure 3: Images synthesized from Brodatz textures

Image	**True** τ	τ_{GL}	$\lvert\tau_{GL}-\tau\rvert$	τ_{HLC}	$\lvert\tau_{HLC}-\tau\rvert$	**True** σ	σ_{GL}	$\lvert\sigma_{GL}-\sigma\rvert$	σ_{HLC}	$\lvert\sigma_{HLC}-\sigma\rvert$
D20	160	160.45	0.45	160.05	0.05	37	37.36	0.36	37.53	0.53
D52	-60	-60.28	0.28	-61.08	1.08	50	49.55	0.45	49.58	0.42
D57	90	91.00	1.00	91.41	1.41	70	70.37	0.37	67.30	2.70
D65	60	59.48	0.52	55.00	5.00	50	48.71	1.29	54.38	4.38
D82	90	89.63	0.37	90.71	0.71	50	49.16	0.84	48.62	1.38
D84	135	134.16	0.84	128.58	6.42	35	35.32	0.32	33.26	1.74
D95	-155	-154.92	0.08	-158.57	3.57	27	25.89	1.11	28.45	1.45

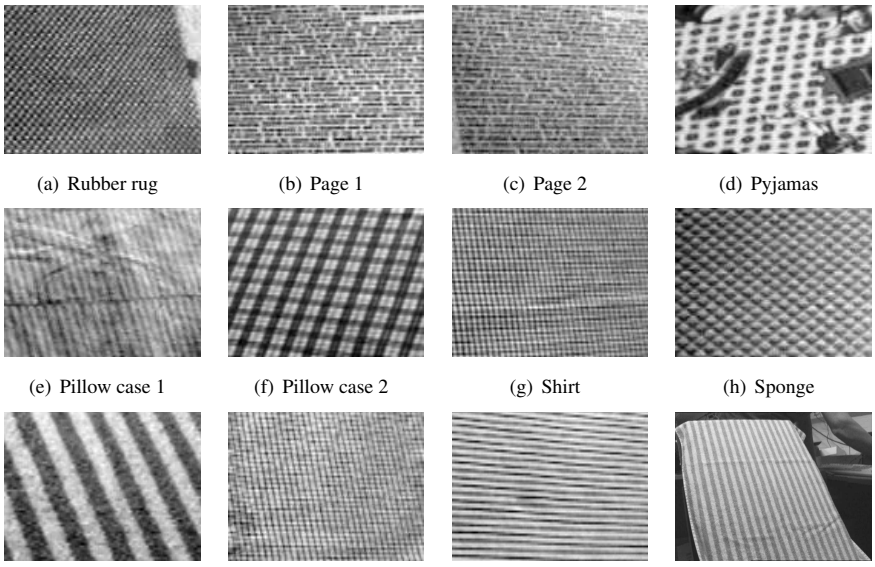
Table 1: Tilt and slant results of our method (τ_{GL}, σ_{GL}) on images synthesized from Brodatz database textures, compared to the results of [8] (τ_{HLC}, σ_{HLC}) (angles in degrees)

All the relevant frequencies are used. Eventually, we choose the pair (σ,τ) with the lowest $V_{\sigma,\tau}$ as we assume that lower values of residual variance, closer to the ideal zero value, correspond to better orientation estimates. As results from all ridges are accurate, future work might address combining these to produce better estimates.

Finally, the algorithm lends itself well to parallel implementations, because each ridge and filter can be processed independently and implemented by different units.

5 Results

We demonstrate our method on two sets of images. The first (figures 3(a)-(g)) is derived from [8], whose results we use for comparison. The images in this set were synthesized by mapping real textures from the Brodatz database ([2]) onto an inclined surface and then rendering it as a new image. Table 1 shows the results achieved compared with those from [8]. Our average estimation errors for τ and σ are 0.51° and 0.68° respectively, while Hwang et al. ([8]) achieve corresponding values of 2.6° and 1.8°. The accuracy of our method is significantly higher. As in [8], we add various levels of white Gaussian noise (SNR ranging from 20 to -5 dB) to the images of the textures D20, D52, D82, D95



(a) Rubber rug (b) Page 1 (c) Page 2 (d) Pyjamas

(e) Pillow case 1 (f) Pillow case 2 (g) Shirt (h) Sponge

(i) Towel (j) Trousers (k) T-shirt (l) Towel in 4(i) - Whole picture

Figure 4: Real images of texture planes

(the latter with SNR=$-$5dB is shown in figure 3(h)). Note that our estimates are always closer to the noiseless result than those of [8], thus indicating increased robustness.

The second set (figures 4(a)-(k)) consists of real images. All of them are the central 128x128 parts of 640x480 pictures. Figure 4(l) presents the whole image from which figure 4(i) was derived. As can be seen, the textured object was laid flat on a panel of known orientation (obtained using a multiple camera system prior to the experiment) and photographed with a Pulnix TM-6EX camera. The chosen textured objects were mainly fabrics, but also included some different materials. It is clear that the pictures are affected by variations in illumination and self shadowing (4(h)), creases (4(e)), imperfections (4(b), 4(a)) and occlusions (4(d)). Table 3 shows the results we obtained, compared to the ground truth. On average, tilt and slant were estimated with an error of 1.3° and 1.5°

	SNR (dB)				
Image (τ/σ)	∞	20	10	0	-5
D20 (160/37)	160.4/37.4	159.5/37.1	159.6/36.8	159.6/37.3	157.3/37.3
D52 (-60/50)	-60.3/49.5	-58.7/47.6	-64.4/49	-67.2/46.6	-61.6/34.3
D82 (90/50)	89.6/49.2	-89.1/51.3	90.6/52.3	86.1/45.4	X
D95 (-155/27)	-154.9/25.9	-158.8/25.2	-159.8/26.2	-160.8/25.4	-160.6/24.5

Table 2: Surface orientations (τ/σ) estimated using our method on noisy images - true values are in parenthesis (X indicates that the results were not reliable) (angles in degrees)

Image	**True** τ	τ_{GL}	$\lvert\tau_{GL}-\tau\rvert$	**True** σ	σ_{GL}	$\lvert\sigma_{GL}-\sigma\rvert$
rubber	118.8	118.3	0.5	35.3	33.4	1.9
page1	-152.8	-153.3	0.5	23.6	23.4	0.2
page2	123.6	121.3	2.3	36.9	34.0	2.9
pyjamas	-152.8	-151.2	1.6	23.6	20.2	3.4
p'case1	-123.6	123.2	0.4	36.9	34.4	2.5
p'case2	-146.5	-147.7	1.2	32.7	33.6	0.9
shirt	103.2	107.5	4.3	33.6	31.2	2.4
sponge	-158.3	-157.9	0.4	25.5	25.1	0.4
towel	146.4	146.2	0.2	38.8	39.9	1.1
trousers	118.8	118.7	0.1	35.3	35.2	0.1
T-shirt	123.6	121.3	2.3	36.9	35.7	1.2

Table 3: Tilt and slant results of our method (τ_{GL},σ_{GL}) on real images (angles in degrees)

respectively. These data confirm both the accuracy and the robustness of our algorithm.

All processed images were 128x128 pixels with 256 levels of gray. The backprojection of the computed LSFs for each value (σ,τ) was done just for the middle section of the image (here 64x64), so as to avoid edge effects. The constant fractional bandwidth is one half, and the space constant of the post-smoothing Gaussian filter is 1/12 of the image. We could not apply our method to those images in [12] because we could not gather all the data of the original setup. Processing the 18 images, 46 ridges of the Fourier transform were detected, that determined 232 Gabor functions. On average the number of convolutions per image was therefore 77.33. Compared to [12], where 378 convolutions per image are used, we save 79.54% in computational power.

As stated in section 1, the homogeneity assumption requires some sort of periodicity/stationarity: the algorithm can deal with as little as 6 cycles/picture.

Finally, we address the possibility that ridges might superimpose. This may be the case when a texture composed of close frequencies is slanted. Such a superposition can easily be spotted by our algorithm, as it results in gaps in the frequency estimation. We solve it by considering a smaller patch of the image, e.g. 96x96 instead of 128x128. In this way the range of variation of frequencies analyzed by the Fourier transform is smaller and hence there is less chance of observing the superposition.

6 Conclusions

The study presented here has characterized the variations of the dominant LSFs in textures via the ridges of their Fourier transforms, and used those to estimate the orientation of surface textures. Numerical results have been given on both semi-synthetic and real images and compared where possible with other work. Our algorithm is more accurate, simple to implement, and has the potential to be extended to more complex surface shapes.

To our knowledge, the proposed algorithm is the first to consider the multi-scale nature of texture to the extent of exploiting all main LSFs. Furthermore, it is robust against shading, variations in illuminations, and occlusions, and performs well in the presence of added Gaussian noise. Finally, it is based on the Fourier transform of the image and on a

minimal number of convolutions, results are therefore computationally fast.

Acknowledgements

The authors thank Dr. B. Super, and Dr W.-L. Hwang, Dr C.-S. Lu, Dr P.-C. Chung for providing texture images. Fabio Galasso is grateful to St.John's college and Giovanni Aldobrandini for the 'A Luisa Aldobrandini' Scholarship, supporting his graduate studies.

References

[1] Alan C. Bovik, Joseph P. Havlicek, and Mita D. Desai. Theorems for discrete filtered modulated signals. In *ICASSP*, pages 805–810, Minneapolis, MN, 1993.

[2] Phil Brodatz. *Textures: A Photographic Album for Artists and Designers*. Dover, New York, 1966.

[3] Maureen Clerc and Stéphane Mallat. Shape from texture through deformations. In *Int. Conf. Comp. Vision*, pages 405–410, 1999.

[4] Maureen Clerc and Stéphane Mallat. The texture gradient equation for recovering shape from texture. *IEEE Trans. Patt. Anal. Mach. Intell.*, 24(4):536–549, 2002.

[5] Antonio Criminisi and Andrew Zisserman. Shape from texture: Homogeneity revisited. In *BMVC*, pages 82–91, 2000.

[6] Jonas Gårding. Shape from texture for smooth curved surfaces in perspective projection. *JMIV*, 2:327–350, 1992.

[7] James J. Gibson. *The Perception of the Visual World*. Houghton Mifflin, Boston, Massachusetts, 1950.

[8] Wen-Liang Hwang, Chun-Shien Lu, and Pau-Choo Chung. Shape from texture: Estimation of planar surface orientation through the ridge surfaces of continuous wavelet transform. *IEEE Trans. Image Processing*, 7(5):773–780, May 1998.

[9] Kenichi Kanatani and Tsai-Chia Chou. Shape from texture: general principle. *Artificial Intelligence*, 38(1):1–48, 1989.

[10] Angeline M. Loh and Richard Hartley. Shape from non-homogeneous, non-stationary, anisotropic, perspective texture. In *BMVC*, pages 69–78, 2005.

[11] Jitendra Malik and Ruth Rosenholtz. Computing local surface orientation and shape from texture for curved surfaces. *IJCV*, 23(2):149–168, 1997.

[12] Boaz J. Super and Alan C. Bovik. Planar surface orientation from texture spatial frequencies. *Pattern Recognition*, 28(5):729–743, 1995.

[13] Boaz J. Super and Alan C. Bovik. Shape from texture using local spectral moments. *IEEE Trans. Patt. Anal. Mach. Intell.*, 17(4):333–343, 1995.

[14] Andrew P. Witkin. Recovering surface shape and orientation from texture. *Artificial Intelligence*, 17:17–45, 1981.

MOTEXATION: Multi-Object Tracking with the Expectation-Maximization Algorithm

Yonggang Jin and Farzin Mokhtarian
Centre for Vision, Speech and Signal Processing
University of Surrey, Guildford, GU2 7XH, UK
{y.jin, f.mokhtarian}@surrey.ac.uk

Abstract

The paper proposes a new edge-based multi-object tracking framework, MOTEXATION, which deals with tracking multiple objects with occlusions using the Expectation-Maximization (EM) algorithm and a novel edge-based appearance model. In the edge-based appearance model, an object is modelled by a mixture of a non-parametric contour model and a non-parametric edge model using kernel density estimation. Visual tracking is formulated as a Bayesian incomplete data problem, where measurements in an image are associated with a generative model which is a mixture of mixture models including object models and a clutter model and unobservable associations of measurements to densities in the generative model are regarded as missing data. A likelihood for tracking multiple objects jointly with an exclusion principle is presented, in which it is assumed that one measurement can only be generated from one density and one density can generate multiple measurements. Based on the formulation, a new probabilistic framework of multi-object tracking with the EM algorithm (MOTEXATION) is presented. Experimental results in challenging sequences demonstrate the robust performance of the proposed method.

1 Introduction

Visual tracking is an important research area of computer vision. Previous work on edge-based contour tracking includes contour tracking with Kalman filtering [3] or particle filtering [9], contour tracking with the EM algorithm [14], which are all for single object tracking. Some similar previous work on joint tracking of multiple objects was presented in [12, 10, 17]. In [10, 17] multi-object tracking with particle filtering was proposed. However the number of samples will grow exponentially with the number of objects, and usually the depth order of multiple objects is needed or needs to be jointly estimated. In [12] Joint Probabilistic Data Association (JPDA) with the exclusion principle is applied to multiple contour tracking in comparison with Probabilistic Data Association (PDA) for single contour tracking in CONDENSATION [9]. Due to the complexity of enumerating all feasible events, the extension to track more than two objects is computationally expensive and also the depth order needs to be estimated and used in the likelihood. On the other hand, many iterative algorithms were proposed for color-based tracking(though only for single object tracking), including mean-shift algorithm with color histogram [6],

kernel-based tracking with spatial-color non-parametric model [8], EM-like tracking with spatial-color Gaussian mixture model [18].

This paper proposes a new edge-based multi-object tracking framework, MOTEXATION, which deals with tracking multiple objects with occlusions using the EM algorithm and a novel edge-based appearance model. The proposed approach differs from previous similar work on contour tracking [3, 9, 12] mainly in three aspects: object model, likelihood and inference used. In the edge-based appearance model, an object is modelled by a mixture of a non-parametric contour model and a non-parametric edge model using kernel density estimation similar to that for color-based non-parametric model [8]. Visual tracking is formulated as a Bayesian incomplete data problem where measurements in an image are associated with a generative model which is a mixture of mixture models including object models and a clutter model and unobservable associations of measurements to densities in the generative model are regarded as missing data. A likelihood for tracking multiple objects jointly with an exclusion principle is presented where it is assumed that 1. one measurement can only be generated from one density 2. one density can generate multiple measurements. The first assumption incorporates the same exclusion principle essential to track objects during occlusion as that of [12], based on JPDA, whereas the second assumption is relaxed like that of Probabilistic Multi-Hypothesis Tracker (PMHT) [15] to allow one density to generate multiple measurements rather than one measurement only. This significantly reduced the complexity of enumerating all feasible events in comparison with JPDA. Tracking multiple objects jointly will increase the dimensionality of state space and often the likelihood will become sharply peaked [16], which makes tracking with particle filtering difficult. The iterative EM algorithm is employed for multi-object tracking due to its monotonicity property which can seek the mode of the likelihood or the posterior despite high dimensional state space and sharply peaked likelihood. In addition it is also possible to combine edge features with color features using the iterative algorithm, for more robust tracking.

The organization of the paper is as follows. Tracking is formulated in Sec. 2; Multi-object tracking with the EM algorithm is presented in Sec. 3; Results are given in Sec. 4 and the paper is concluded in Sec. 5.

2 Tracking formulation

State vector is denoted as $\mathbf{x}(t) = [x(t)\, y(t)\, a(t)\, b(t)]^T$ where $[x(t)\, y(t)]^T$ is the spatial position of the object centre, $a(t)$ and $b(t)$ are the width and height of the object respectively. A second order auto-regressive model is employed as the dynamical model, $\mathbf{x}(t) = \mathbf{A}_1\mathbf{x}(t-1) + \mathbf{A}_2\mathbf{x}(t-2) + \mathbf{B_0}\mathbf{w}(t)$ where $\mathbf{w}(t)$ is Gaussian noise $\mathcal{N}(\mathbf{w}(t); \mathbf{0}, \mathbf{I})$.

2.1 Gating and clustering

Edge measurements are first detected by Canny edge detector [5]. The gating procedure of PDA is then applied. A validation region is computed based on the predicted state vector using dynamical model for each object so only measurements from within the validation region of the predicted state vector are used [1].

The clustering procedure from JPDA is also employed [1] for multi-object tracking. Multiple objects are first grouped into clusters and then are tracked jointly in each cluster. It often occurs that more than one object are grouped into the same cluster if there are occlusions between objects. After clustering, measurements in validation regions of all

objects in a cluster are used for jointly tracking multiple objects in that cluster. Measurements in a cluster are denoted as $\mathbf{Z} = \{\mathbf{z}_i\}_{i=1}^{N}$, where N is the number of measurements in a cluster, $\mathbf{z}_i = \begin{bmatrix} \mathbf{u}_i \\ \mathbf{v}_i \end{bmatrix}$, $\mathbf{u}_i = [x_i, y_i]^T$ and $\mathbf{v}_i = \theta_i \in [0, 2\pi)$ are the spatial position and orientation of ith edge measurement respectively.

2.2 Object model

The edge-based object appearance model $p_l(\mathbf{z})$ is a mixture of a non-parametric contour model $p_{con}(\mathbf{z})$, which consists of contour sample points, and a non-parametric edge model $p_{edge}(\mathbf{z})$, which consists of edge pixels inside the object contour, $p_l(\mathbf{z}) = \pi_{con} p_{con}(\mathbf{z}) + \pi_{edge} p_{edge}(\mathbf{z})$ where π_{con} and π_{edge} is the mixture weight of contour model and edge model respectively, $\pi_{con} + \pi_{edge} = 1$.

For the non-parametric contour model,

$$p_{con}(\mathbf{z}) = \frac{1}{M_{con}} \sum_{j=1}^{M_{con}} \mathscr{K}_{con}(\mathbf{z}; \mathbf{m}_{con,j}, \Sigma) = \frac{1}{M_{con}} \sum_{j=1}^{M_{con}} \mathscr{N}(\mathbf{u}; \mathbf{u}_{con,j}, \Sigma_{\mathbf{u}}) \mathscr{K}_{\mathbf{v},con}(\mathbf{v}; \mathbf{v}_{con,j}, \Sigma_{\mathbf{v}})$$

where $\mathbf{m}_{con,j} = \begin{bmatrix} \mathbf{u}_{con,j} \\ \mathbf{v}_{con,j} \end{bmatrix}$, $\mathbf{u}_{con,j}$ and $\mathbf{v}_{con,j} = \theta_{con,j} \in [0, \pi)$ are the spatial position and orientation of the normal of jth contour sample respectively, $\Sigma = \begin{bmatrix} \Sigma_{\mathbf{u}} & \mathbf{0} \\ \mathbf{0} & \Sigma_{\mathbf{v}} \end{bmatrix}$, $\Sigma_{\mathbf{u}}$ and $\Sigma_{\mathbf{v}} = \sigma_\theta^2$ are the fixed covariance of spatial position and orientation respectively, $\mathscr{K}_{\mathbf{v},con}(\mathbf{v}; \mathbf{v}_{con,j}, \Sigma_{\mathbf{v}}) \propto e^{-\frac{d_{con}^2(\theta, \theta_{con,j})}{2\sigma_\theta^2}}$ and $d_{con}(\theta, \theta_{con,j}) \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$. Object contour is expressed parametrically by $\mathbf{m}_{con} = f(s, \mathbf{x})$ where s is the contour parameter. An ellipse can be used for head tracking and more complex contours can be represented by B-spline [4].

For the non-parametric edge model,

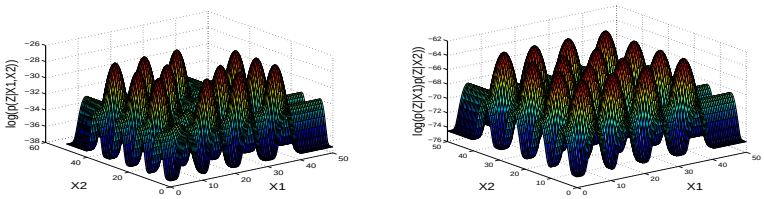
$$p_{edge}(\mathbf{z}) = \frac{1}{M_{edge}} \sum_{j=1}^{M_{edge}} \mathscr{K}_{edge}(\mathbf{z}; \mathbf{m}_{edge,j}, \Sigma) = \frac{1}{M_{edge}} \sum_{j=1}^{M_{edge}} \mathscr{N}(\mathbf{u}; \mathbf{u}_{edge,j}, \Sigma_{\mathbf{u}}) \mathscr{K}_{\mathbf{v},edge}(\mathbf{v}; \mathbf{v}_{edge,j}, \Sigma_{\mathbf{v}})$$

where $\mathbf{m}_{edge,j} = \begin{bmatrix} \mathbf{u}_{edge,j} \\ \mathbf{v}_{edge,j} \end{bmatrix}$, $\mathbf{u}_{edge,j}$ and $\mathbf{v}_{edge,j} = \theta_{edge,j} \in [0, 2\pi)$ are the spatial position and orientation of jth edge pixel inside the object contour respectively, $\mathscr{K}_{\mathbf{v},edge}(\mathbf{v}; \mathbf{v}_{edge,j}, \Sigma_{\mathbf{v}})$ $\propto e^{-\frac{d_{edge}^2(\theta, \theta_{edge,j})}{2\sigma_\theta^2}}$ and $d_{edge}(\theta, \theta_{edge,j}) \in [-\pi, \pi]$.

Note that contour model $p_{con}(\mathbf{z})$ can be regarded as a "stable" component and edge model $p_{edge}(\mathbf{z})$ as a "wandering" component in the object model [11]. Rewrite $p_l(\mathbf{z})$ as $p_l(\mathbf{z}) = \sum_{j=1}^{M} \omega_j \mathscr{N}(\mathbf{u}; \mathbf{u}_j, \Sigma_{\mathbf{u}}) \mathscr{K}_{\mathbf{v},j}(\mathbf{v}; \mathbf{v}_j, \Sigma_{\mathbf{v}})$ where $\{\omega_j\}_{j=1}^{M} = \left\{ \left\{ \frac{\pi_{con}}{M_{con}} \right\}_{j=1}^{M_{con}}, \left\{ \frac{\pi_{edge}}{M_{edge}} \right\}_{j=1}^{M_{edge}} \right\}$, $\{\mathbf{m}_j\}_{j=1}^{M} = \left\{ \{\mathbf{m}_{con,j}\}_{j=1}^{M_{con}}, \{\mathbf{m}_{edge,j}\}_{j=1}^{M_{edge}} \right\}$, $M = M_{con} + M_{edge}$ and later on for brevity, it will not be specified whether a density is from contour model or edge model.

2.3 Clutter model

A clutter model $p_c(\mathbf{z})$ is used to assimilate the measurements not from objects. It also corresponds to a "lost" component [11]. Uniform density is used so $p_c(\mathbf{z}) = p_c = \frac{1}{V_{\mathbf{u}} \times V_{\mathbf{v}}}$



(a) (b)

Figure 1: Comparison of (a) joint tracking likelihood $p(\mathbf{Z}|\mathbf{x}_1,\mathbf{x}_2)$ with exclusion principle and (b) separate tracking likelihood $p(\mathbf{Z}|\mathbf{x}_1)p(\mathbf{Z}|\mathbf{x}_2)$.

where $V_{\mathbf{u}}$ and $V_{\mathbf{v}}$ are the volume of validation region and of feature space without validation respectively [1].

2.4 Likelihoods

To explain measurements of a cluster with more than one object, the generative model is a mixture of mixture models including transformed mixture models of all objects in that cluster and a clutter model. The generative model can be written as $p(\mathbf{z}|\mathbf{x}) = \pi_c p_c(\mathbf{z}) + \sum_{l=1}^{L} \pi_l p_l(\mathbf{z}|\mathbf{x}_l)$, where $\mathbf{x} = \{\mathbf{x}_l\}_{l=1}^{L}$ includes state vectors of L objects in a cluster, π_l and π_c are the mixture weight of the lth object model and clutter model respectively and $\pi_c + \sum_{l=1}^{L} \pi_l = 1$, $p_l(\mathbf{z}|\mathbf{x}_l) = \sum_{j=1}^{M_l} \omega_{l,j} \mathcal{N}(\mathbf{u}; T_{\mathbf{u}}(\mathbf{u}_{l,j}, \mathbf{x}_l), \Sigma_{\mathbf{u}}) \mathcal{K}_{\mathbf{v},l,j}(\mathbf{v}; \mathbf{v}_{l,j}, \Sigma_{\mathbf{v}})$ is the transformed lth object model assuming unchanged orientation feature vector, M_l and $\omega_{l,j}$ are the number of densities and jth mixture weight in the lth object model respectively.

Assuming measurements $\mathbf{Z}$ are drawn independently from the generative model $p(\mathbf{z}|\mathbf{x})$, the likelihood given the incomplete data $\mathbf{Z}$ is

$$p(\mathbf{Z}|\mathbf{x}) = \prod_{i=1}^{N} p(\mathbf{z}_i|\mathbf{x}) = \prod_{i=1}^{N} \left[\pi_c p_c + \sum_{l=1}^{L} \pi_l p_l(\mathbf{z}_i|\mathbf{x}_l) \right] \tag{1}$$

Despite its simplicity, the same exclusion principle as that in [12] is included in the likelihood 1 in comparison with likelihood of tracking multiple objects separately

$$\mathscr{L}(\mathbf{x}) = \prod_{l=1}^{L} p(\mathbf{Z}|\mathbf{x}_l) = \prod_{l=1}^{L} \prod_{i=1}^{N} [\pi_c p_c + (1.0 - \pi_c) p(\mathbf{z}_i|\mathbf{x}_l)] \tag{2}$$

Fig. 1 illustrates a 1D example with 4 measurements and 2 objects with 1 density each as that in [12].

In practice the assumption of independent measurements is not valid if measurements are close to each other as there are strong correlations between measurements [16]. A more practical likelihood is to incorporate measurement weights described in section 2.5,

$$p(\mathbf{Z}|\mathbf{x}) = \prod_{i=1}^{N} \left[\pi_c p_c + \sum_{l=1}^{L} \pi_l p_l(\mathbf{z}_i|\mathbf{x}_l) \right]^{\alpha_i} \tag{3}$$

where α_i is weight for ith measurement.

From the viewpoint of the Bayesian incomplete data problem, the missing data of association of measurements with densities are introduced and denoted as $\mathbf{K} = \{\mathbf{k}_i\}_{i=1}^{N}$ and $\mathbf{k}_i = \{k_i^1, k_i^2\}$ where $k_i^1 \in \{1, \cdots, L, c\}$, $k_i^1 = c$ indicates the association with clutter, and $k_i^1 = l, l \in \{1, \cdots, L\}$ association with object l; $k_i^2 \in \{1, \cdots M_{k_i^1}\}$ gives the association with one of the mixture densities in k_i^1th model. Assuming that 1. a measurement can have only one source 2. more than one measurement can originate from a density, where the first assumption is the same as that of JPDA known as exclusion principle in [12] and the second assumption is relaxed similar to that of PMHT, there are $N_e = (\sum_{l=1}^{L} M_l + 1)^N$ feasible events $\{\chi_n\}_{n=1}^{N_e}$. The likelihood given the complete data is

$$p(\mathbf{Z}, \mathbf{K} = \mathbf{K}(\chi_n)|\mathbf{x}) \propto \prod_{i:k_i^1(\chi_n)=c} \pi_c p_c \prod_{i,l,j:\substack{k_i^1(\chi_n)=l \\ k_i^2(\chi_n)=j}} \pi_l \omega_{l,j} \mathcal{N}(\mathbf{u}_i; T_{\mathbf{u}}(\mathbf{u}_{l,j}, \mathbf{x}_l), \Sigma_{\mathbf{u}}) \mathcal{K}_{\mathbf{v},l,j}(\mathbf{v}_i; \mathbf{v}_{l,j}, \Sigma_{\mathbf{v}}) \tag{4}$$

For comparison, JPDA can also be viewed in light of Bayesian incomplete data problem with a slightly different assumption that 1. a measurement can have only one source 2. no more than one measurement can originate from a density, so there are $\sum_{n=0}^{\min(N,M)} \frac{M!N!}{(M-n)!(N-n)!n!}$ feasible events. Denote $N_0(\chi_n)$ as number of densities which have no allocated measurements and $N_1(\chi_n)$ as number of densities which have only one allocated measurement in a feasible event χ_n, the likelihood given complete data in JPDA is

$$p(\mathbf{Z}, \mathbf{K} = \mathbf{K}(\chi_n)|\mathbf{x}) \propto {p_c}^{N-N_1(\chi_n)} \mu_F(N - N_1(\chi_n))(1 - P_D P_G)^{N_0(\chi_n)} (P_D)^{N_1(\chi_n)} \frac{(N-N_1(\chi_n))!}{N!}$$
$$\times \prod_{i,l,j:\substack{k_i^1(\chi_n)=l \\ k_i^2(\chi_n)=j}} \mathcal{N}(\mathbf{u}_i; T_{\mathbf{u}}(\mathbf{u}_{l,j}, \mathbf{x}_l), \Sigma_{\mathbf{u}}) \mathcal{K}_{\mathbf{v},l,j}(\mathbf{v}_i; \mathbf{v}_{l,j}, \Sigma_{\mathbf{v}})$$

where P_D is the detection probability, P_G is the probability that the true measurement will fall in the validation region, $\mu_F(n)$ is the probability mass function of the number of false measurements [1].

After marginalization of equation (4), $p(\mathbf{Z}|\mathbf{x})$ is factorized to N terms in equation (1) in comparison with $\sum_{n=0}^{\min(N,M)} \frac{M!N!}{(M-n)!(N-n)!n!} \gg N$ terms in marginalized likelihood of JPDA.

2.5 Measurement weighting

Histogram back-projection is used to incorporate background information. A background edge orientation histogram $\{h_i\}_{i=1}^{N_B}$ with N_B bins of orientation is built by using the edge pixels in a rectangular window surrounding each object. The background histogram is adapted online by weighted sum of previous background histogram and background histogram built given current object state estimation.

A ratio histogram $\{r_i\}_{i=1}^{N_B}$ is computed by $r_i = \min\left(\frac{\hat{h}}{h_i}, 1\right)$ where $\hat{h} = \min_{i:h_i>0}(h_i)$. Measurement weight α_i is computed from the ratio histogram as $\alpha_i = \frac{r_{b(\mathbf{z}_i)}}{\sum_{i=1}^{N} r_{b(\mathbf{z}_i)}} \times \frac{1}{2\sigma^2}$ where $b(\mathbf{z}_i)$ denotes the bin to which $\mathbf{z}_i$ belongs and σ is a constant.

(a) Initial estimation (b) Final estimation (c) Lower bound

Figure 2: Iterative update of EM algorithm where edge measurements are marked in yellow: (a) initial estimation, (b) final estimation, (c) lower bound increasing monotonically.

Measurements with orientations occurring most commonly in the background will have the lowest weight and measurements with orientations which are not in the background will have the highest weight. If the ratio histogram is uniform, it degenerates to the case that each measurement has the same weight $\alpha_i = \frac{1}{N} \times \frac{1}{2\sigma^2}$.

3 Multi-object tracking with the EM algorithm

State vector $\mathbf{x}(t)$ is estimated by either Maximum Likelihood (ML) estimation $\hat{\mathbf{x}}(t) = \arg\max_{\mathbf{x}(t)} p(\mathbf{Z}(t)|\mathbf{x}(t))$ or Maximum a Posteriori (MAP) estimation $\hat{\mathbf{x}}(t) = \arg\max_{\mathbf{x}(t)} p(\mathbf{x}(t)|\mathscr{Z}(t))$, where $\mathscr{Z}(t) = \{\mathbf{Z}(j)\}_{j=0}^{t}$, using the EM algorithm [7] and its generalization [13].

From Jensen's inequality it can be shown that

$$\log p(\mathbf{Z}|\mathbf{x}) = \sum_{i=1}^{N} \alpha_i \log \left[\frac{\pi_c p_c(\mathbf{z}_i)}{q_{i,c}} q_{i,c} + \sum_{l=1}^{L} \sum_{j=1}^{M_l} q_{i,l,j} \frac{\pi_l \omega_{l,j} \mathscr{N}(\mathbf{u}_i; T_{\mathbf{u}}(\mathbf{u}_{l,j}, \mathbf{x}_l), \Sigma_{\mathbf{u}}) \mathscr{K}_{\mathbf{v},l,j}(\mathbf{v}_i; \mathbf{v}_{l,j}, \Sigma_{\mathbf{v}})}{q_{i,l,j}} \right]$$
$$\geq \sum_{i=1}^{N} \alpha_i \left[q_{i,c} \log \frac{\pi_c p_c(\mathbf{z}_i)}{q_{i,c}} + \sum_{l=1}^{L} \sum_{j=1}^{M_l} q_{i,l,j} \log \frac{\pi_l \omega_{l,j} \mathscr{N}(\mathbf{u}_i; T_{\mathbf{u}}(\mathbf{u}_{l,j}, \mathbf{x}_l), \Sigma_{\mathbf{u}}) \mathscr{K}_{\mathbf{v},l,j}(\mathbf{v}_i; \mathbf{v}_{l,j}, \Sigma_{\mathbf{v}})}{q_{i,l,j}} \right]$$

where $q_{i,c} = p(k_i^1 = c)$, $q_{i,l,j} = p(k_i^1 = l, k_i^2 = j)$ are the probabilities of missing data $\mathbf{K}$ and $q_{i,c} + \sum_{l=1}^{L} \sum_{j=1}^{M_l} q_{i,l,j} = 1$. So the lower bound of likelihood $J_{ML}(\mathbf{Q}, \mathbf{x}(t))$ for ML estimation and lower bound of posterior $J_{MAP}(\mathbf{Q}, \mathbf{x}(t))$ for MAP estimation are

$$J_{ML}(\mathbf{Q}, \mathbf{x}(t)) = \sum_{i=1}^{N} \alpha_i \left[q_{i,c} \log \frac{\pi_c p_c(\mathbf{z}_i)}{q_{i,c}} + \sum_{l=1}^{L} \sum_{j=1}^{M_l} q_{i,l,j} \log \frac{\pi_l \omega_{l,j} \mathscr{N}(\mathbf{u}_i; T_{\mathbf{u}}(\mathbf{u}_{l,j}, \mathbf{x}_l(t)), \Sigma_{\mathbf{u}}) \mathscr{K}_{\mathbf{v},l,j}(\mathbf{v}_i; \mathbf{v}_{l,j}, \Sigma_{\mathbf{v}})}{q_{i,l,j}} \right] \tag{5}$$

$$J_{MAP}(\mathbf{Q}, \mathbf{x}(t)) = J_{ML}(\mathbf{Q}, \mathbf{x}(t)) + \log p(\mathbf{x}(t)|\mathscr{Z}(t-1)) \tag{6}$$

where $\mathbf{Q} = \left\{ q_{i,c}, \left\{ \{q_{i,l,j}\}_{j=1}^{M_l} \right\}_{l=1}^{L} \right\}_{i=1}^{N}$.

The prior is given by

$$p(\mathbf{x}(t)|\mathscr{Z}(t-1)) = \prod_{l=1}^{L} p(\mathbf{x}_l(t)|\mathscr{Z}(t-1)) = \prod_{l=1}^{L} \mathscr{N}(\mathbf{x}_l(t); \tilde{\mathbf{x}}_l(t), \tilde{\mathbf{P}}_l(t)) \tag{7}$$

Algorithm 1 Multi-Object Tracking with the EM Algorithm (MOTEXATION)

1. Predict by equation (7)
2. EM algorithm
   $k = 1$, $\mathbf{x}^{(0)}(t) = \tilde{\mathbf{x}}(t)$
   (i) E-step by equation (8)
   (ii) M-step by equation (9) or equation (10)
   if $\left\| \mathbf{x}_l^{(k)}(t) - \mathbf{x}_l^{(k-1)}(t) \right\| < \varepsilon, l = 1, \cdots, L$ **then**
   $\hat{\mathbf{x}}(t) = \mathbf{x}^{(k)}(t)$ and stop
   else
   $k = k+1$ go to (i)
   end if

where $\tilde{\mathbf{x}}_l(t) = \mathbf{A}_1\hat{\mathbf{x}}_l(t-1) + \mathbf{A}_2\hat{\mathbf{x}}_l(t-2)$ and $\tilde{\mathbf{P}}_l(t) \approx \mathbf{B}_0\mathbf{B}_0^T$ are the predicted state vector and covariance of lth object respectively,

In E-step, given fixed $\mathbf{x}^{(k-1)}(t)$, maximize $J_{ML}(\mathbf{Q}, \mathbf{x}(t))$ or $J_{MAP}(\mathbf{Q}, \mathbf{x}(t))$. Let $T_{\mathbf{u}}(\mathbf{u}_{l,j}, \mathbf{x}_l(t))$ $= \mathbf{W}_{l,j}\mathbf{x}_l(t)$ where $\mathbf{W}_{l,j}$ is Jacobian of the transformation. At iteration k, $\mathbf{Q}^{(k)}$ is

$$\begin{aligned} q_{i,c}^{(k)} &\propto \pi_c p_c(\mathbf{z}_i) \\ q_{i,l,j}^{(k)} &\propto \pi_l \omega_{l,j} \mathcal{N}(\mathbf{u}_i; \mathbf{W}_{l,j}\mathbf{x}_l^{(k-1)}(t), \Sigma_{\mathbf{u}}) \mathcal{K}_{\mathbf{v},l,j}(\mathbf{v}_i; \mathbf{v}_{l,j}, \Sigma_{\mathbf{v}}) \\ q_{i,c}^{(k)} &+ \sum_{l=1}^{L}\sum_{j=1}^{M_l} q_{i,l,j}^{(k)} = 1, i = 1 \cdots N \end{aligned} \tag{8}$$

In M-step, given $\mathbf{Q}^{(k)}$, maximize $J_{ML}(\mathbf{Q}, \mathbf{x}(t))$ or $J_{MAP}(\mathbf{Q}, \mathbf{x}(t))$. At iteration k, $\mathbf{x}(t)$ is given by

$$\mathbf{x}_{l,ML}^{(k)}(t) = \left[\sum_{j=1}^{M_l} \mathbf{W}_{l,j}^T \tilde{\Sigma}_{l,j}^{(k)^{-1}} \mathbf{W}_{l,j} \right]^{-1} \left[\sum_{j=1}^{M_l} \mathbf{W}_{l,j}^T \tilde{\Sigma}_{l,j}^{(k)^{-1}} \tilde{\mathbf{u}}_{l,j}^{(k)} \right] \tag{9}$$

or

$$\mathbf{x}_{l,MAP}^{(k)}(t) = \left[\sum_{j=1}^{M_l} \mathbf{W}_{l,j}^T \tilde{\Sigma}_{l,j}^{(k)^{-1}} \mathbf{W}_{l,j} + \tilde{\mathbf{P}}_l^{-1}(t) \right]^{-1} \left[\sum_{j=1}^{M_l} \mathbf{W}_{l,j}^T \tilde{\Sigma}_{l,j}^{(k)^{-1}} \tilde{\mathbf{u}}_{l,j}^{(k)} + \tilde{\mathbf{P}}_l^{-1}(t) \tilde{\mathbf{x}}_l(t) \right] \tag{10}$$

where $\tilde{\mathbf{u}}_{l,j}^{(k)} = \frac{\sum_{i=1}^{N} \alpha_i q_{i,l,j}^{(k)} \mathbf{u}_i}{\sum_{i=1}^{N} \alpha_i q_{i,l,j}^{(k)}}$ is the synthetic measurement and $\tilde{\Sigma}_{l,j}^{(k)} = \frac{\Sigma_{\mathbf{u}}}{\sum_{i=1}^{N} \alpha_i q_{i,l,j}^{(k)}}$ is the synthetic covariance.

The main stages of multi-object tracking with the EM algorithm are given in algorithm (1) and the iterative update of MAP estimation is shown in Fig. 2 where the lower bound of posterior is also verified to be increased monotonically.

4 Results

The experiments are carried out in challenging test sequences with heavy occlusions. With unfully optimized C++ code, it runs comfortably at average 0.071s per object per frame

t=5210 t=5286 t=5380 t=5420 t=5430

Figure 3: Tracking results of *"office"* sequence. ©Mitsubishi Electric ITE 2005.

t=425 t=433 t=450 t=460 t=475

Figure 4: Tracking results of *"head"* sequence.

t=1060 t=1180 t=1280 t=1296 t=1380

Figure 5: Tracking results of *"Caviar OneShopOneWait2cor"* sequence.

on 3GHz Pentium IV. Note that to illustrate joint tracking of multiple objects in a cluster, white lines show the links between objects which are tracked jointly in the same cluster.

Three results of multiple head tracking are shown and the size of head also varies from small ones to large ones. Fig 3 shows multi-object tracking results on the *"office"* sequence, in which there are dramatic appearance changes, scale changes and four heavy occlusions. The light green ellipse occluded dark green ellipse from frame 5280 to 5320, from frame 5340 to 5370 and from frame 5380 to 5410. The red ellipse occluded both light green and dark green ellipses from frame 5410 to 5424.

The results of *"head"*[1] are then given in Fig. 4 where there are two heavy occlusions from frame 420 to 442 and from frame 452 to 468.

Fig. 5 shows the results of *"Caviar* [2] *OneShopOneWait2cor"* sequence where the size of target heads are quite small and there are two heavy occlusions from frame 1166 to 1176 and from frame 1276 to 1292.

To track more complex contours, a B-spline contour model is learned as that of [2, 4]. Results of *"Caviar EnterExitCrossingPaths1cor2"* sequence are given in Fig 6 where there are large appearance changes, scale changes and one heavy occlusion from frame 86 to 100.

Fig. 7 presents the results of *"Caviar OneStopMoveEnter1cor2"* sequence, a very crowded and cluttered scene involving large appearance changes, scale changes and also one heavy occlusion from frame 256 to 272.

[1] The sequence is from http://vision.stanford.edu/ birch/headtracker/.

[2] The EC Funded CAVIAR project/IST 2001 37540, see http://homepages.inf.ed.ac.uk/rbf/CAVIAR/.

t=0 t=80 t=90 t=150 t=350

Figure 6: Tracking results of *"Caviar EnterExitCrossingPaths1cor2"* sequence.

t=0 t=262 t=276 t=300 t=480

Figure 7: Tracking results of *"Caviar OneStopMoveEnter1cor2"* sequence.

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

Figure 8: Examples of tracking failure. (a)(b) tracking multiple objects separately using the EM algorithm, (c) contour tracking with CONDENSATION, (d) mean-shift tracking with color histogram.

It should be noted that if multiple objects are tracked separately using the EM algorithm with likelihood 2, which does not have exclusion principle, objects may be lost during occlusion as shown in Fig. 8(a)(b). The proposed method has also been compared with contour tracking using CONDENSATION [9], color tracking using mean-shift [6] and both failed when there are heavy occlusions. Examples of tracking failure are shown in Fig. 8(c)(d).

5 Conclusions

The paper proposes a new edge-based multi-object tracking framework, MOTEXATION, which deals with tracking multiple objects with occlusions using the EM algorithm and a novel edge-based appearance model. In the edge-based appearance model, an object is modelled by a mixture of a non-parametric contour model and a non-parametric edge model using kernel density estimation. Visual tracking is formulated as a Bayesian incomplete data problem where measurements in an image are associated with a generative model which is a mixture of mixture models including object models and a clutter model and unobservable associations of measurements to densities in the generative model are regarded as missing data. A likelihood for tracking multiple objects jointly with an exclusion principle is presented. Based on the formulation, a new probabilistic framework

of multi-object tracking with the EM algorithm (MOTEXATION) is presented. Results in challenging sequences demonstrate the robust performance of the proposed method.

Acknowledgements

The support of the Visual Information Laboratory (VIL), Mitsubishi Electric Information Technology Centre Europe (ITE) and Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey is gratefully acknowledged. We would also like to thank Dr O'Callaghan, Dr Bober and Dr Ratliff of VIL for helpful discussions and for providing the *"office"* sequence.

References

[1] Y. Bar-Shalom and T. Fortmann. *Tracking and Data Association*. Academic Press, 1988.

[2] A.M. Baumberg and D.C. Hogg. Learning flexible models from image sequences. In *Proc. ECCV*, pages 299–308, 1994.

[3] A. Blake, R. Curwen, and A. Zisserman. A framework for spatio-temporal control in the tracking of visual contours. *IJCV*, 11(2):127–145, 1993.

[4] A. Blake and M. Isard. *Active Contours*. Springer, 1998.

[5] J. Canny. A computational approach to edge detection. *IEEE PAMI*, 8(6):679–698, Nov. 1986.

[6] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Proc. CVPR*, pages 142–149, 2000.

[7] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximal likelihood from incomplete data via the EM algorithm. *RoyalStat*, B 39:1–38, 1977.

[8] A. Elgammal, R. Duraiswami, and L. Davis. Probabilistic tracking in joint feature-spatial spaces. In *Proc. IEEE CVPR*, pages 781–788, 2003.

[9] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *Proc. ECCV*, pages 343–356, 1996.

[10] M. Isard and J. MacCormick. BraMBLe: a Bayesian multiple-blob tracker. In *Proc. ICCV*, pages 34–41, 2001.

[11] A. D. Jepson, D. J. Fleet, and T. F. Ei-Maraghi. Robust online appearance models for visual tracking. *IEEE PAMI*, 25(10):1296–1311, October 2003.

[12] J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. In *Proc. ICCV*, pages 572–578, 1999.

[13] R. M. Neal and G. E. Hinton. A new view of the EM algorithm that justifies incremental, sparse and other variants. *Learning in Graphical Models*, pages 355–368, 1998.

[14] A.E.C. Pece and A.D. Worrall. Tracking with the EM contour algorithm. In *Proc. ECCV*, pages 3–17, 2002.

[15] R. Streit and T. Luginbuhl. Maximum likelihood method for probabilistic multi-hypothesis tracking. *Proc. SPIE*, 2235:394–405, 1994.

[16] J. Sullivan, A. Blake, M. Isard, and J. MacCormick. Object localization by Bayesian correlation. In *Proc. ICCV*, pages 1068–1075, 1999.

[17] Y. Wu, T. Yu, and G. Hua. Tracking appearances with occlusions. In *Proc. CVPR*, pages 789–795, 2003.

[18] T. Yu and Y. Wu. Differential tracking based on spatial-appearance model (SAM). In *Proc. CVPR*, pages 720–727, 2006.

Multicues 3D Monocular Upper Body Tracking Using Constrained Belief Propagation

Philippe Noriega, Olivier Bernier
France Telecom Research and Development
France Telecom R&D
2 Av. Pierre Marzin 22307 Lannion Cedex France
{philippe.noriega, olivier.bernier}@orange-ftgroup.com

Abstract

This paper describes a method for articulated 3D upper body tracking in monocular scenes using a graphical model to represent an articulated body structure. Belief propagation on factor graphs is used to compute the marginal probabilities of limbs. The body model is a loose-limbed model including attraction factors between adjacent limbs and constraints to reject poses resulting in collisions. To solve ambiguities resulting from monocular view, robust contour and colour based cues are extracted from the images. Moreover, a set of constraints on the model articulations is implemented according to human pose capabilities. Quantitative and qualitative results illustrate the efficiency of the proposed algorithm.

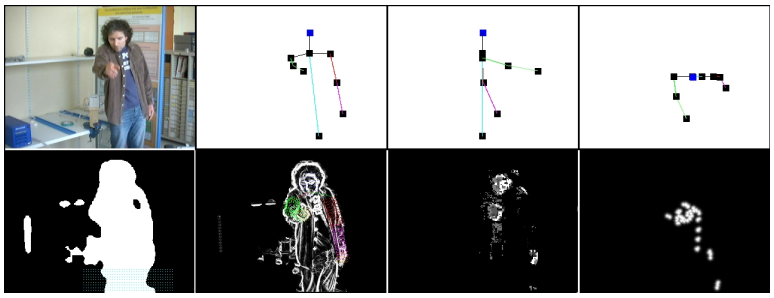


Figure 1: Upper body tracking. First row: original image, front, right side and top views of the obtained limbs positions with a single camera. Second row: background subtraction, contours, face colour map and energy motion distance map.

1 Introduction

Algorithms for body tracking must cope with a high dimensional space in which the joint probability function is highly multimodal and sharp. In this context, deterministic

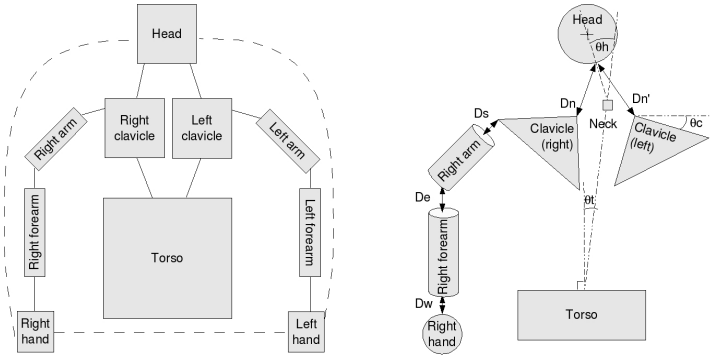


Figure 2: Limbs interactions (Left): nodes correspond to limbs, articulation constraints are represented by solid lines and dashed lines are additional non-collision constraints between head and hands. Upper body model (Right): arms and forearms are modeled by cylinders and the head by a sphere. Other limbs (hands torso and clavicles) are represented by 2D patches. Limb interaction factors are computed with the distances (Dn, Ds, De, Dw) between them. Other joints constraints are determined by the angles θh, θc and θt. The neck is located at equal distance from both clavicles.

methods can track in real time with stereo cameras [5], but may fail for monocular view because of the many local optimums owing to ambiguities in monocular scenes [13].

Due to articulation constraints, consistent poses are bounded in a smaller subspace making learning based tracking methods efficient if their learning set sufficiently covers this subspace. Various regressions methods, aiming at deducing a pose directly from an image, have been tested on walking sequences with constrained environments [2]. Non negative matrix factorisation [1] can enhance such methods by rejecting non discriminative data. Other methods like GPDM [15] introduce probabilities in the computation of a latent space to smooth the resulting pose, but test scenes are restricted to cyclic motions. Other methods that perform a comparison between an image and a learning base require a huge database even when robust locally-weighted regression between candidates poses is used [10]. Increasing the data base may slow down drastically the comparison process and, to speed up the selection of a subset of nearest neighbours, the comparison process can use locally sensitive hashing and Hamming distance [14]. The likelihood of a body pose is computed with this previous method using a Bayesian framework but some poses that are dissimilar to the learned ones are not correctly estimated and generally, the huge pose space and the variability in external parameters such as clothing or hairstyle is the major cause of failure in learning based methods.

Stochastic algorithms are useful in monocular vision to resolve ambiguities resulting from 2D to 3D pose inference, particularly when a multi-hypothesis algorithm, such as particle filtering [4], is used. The main drawback with such methods is the high dimensional pose space. A way to avoid this problem consists in using a loose-limbed body model [11] where the likelihood of each limb is evaluated independently. In this manner, a particle filter can be associated with each limb reducing the search space dimension to the number of *dof* of a limb [3]. Influence between limbs is taken into account by propagating limb beliefs through a factor graph using belief propagation [8]. A similar

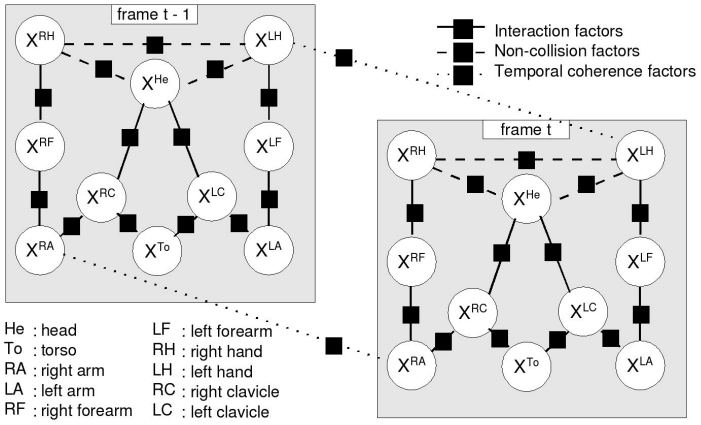


Figure 3: Factor graph. Circles corresponds to variable nodes (limb states) and black squares to factor nodes (temporal coherence T^{μ} and interaction or non-collision factors $\psi^{\mu\nu}$). For clarity, only two consecutive frames with two temporal factor links are shown and the factor nodes corresponding to the observations Y^{μ} are omitted.

technique is used in monocular scenes [7] with only motion energy as cue to detect arms and forearms position.

In this paper, the number of cues is increased to enhance the robustness of the tracking. Moreover, the use of interacting particle filters with belief propagation [3] simplify our algorithm by computing recursively an estimation in a discrete space instead of using, for example, a Gibbs sampler in a continuous one [11]. More general articulation constraints rules are built in the compatibility factors computation instead of learning them from specific walking sequences with a mixture of Gaussians [11]. The proposed algorithm performs at six *fps* using a standard webcam.

2 Recursive Bayesian tracking

The upper body is modeled as a graph including M limbs represented by nodes and links corresponding to articulations or non collision constraints between limbs (figure 2). Basically, a Markov network can be used to represent this structure but the non-collision constraints between the head and the hands generate a three nodes clique. A factor graph is constructed to simplify the model by using only pairwise factors [3]. The joint probability can be decomposed as a products of these factors. The complete graph includes the previous states to take into account the temporal coherence (figure 3). Given a limb μ, its state X_t^{μ} at time t and the image observations Y_t^{μ}, the model parameters are the observations compatibility factors $\phi^{\mu}(X^{\mu},Y^{\mu})$, the time interaction factors $T^{\mu}(X_t^{\mu},X_{t-1}^{\mu})$, and the interaction factor for the link between limbs μ and ν: $\psi^{\mu\nu}(X^{\mu},X^{\nu})$. Adopting these notations, the joint probability knowing all observations from time 0 to T is:

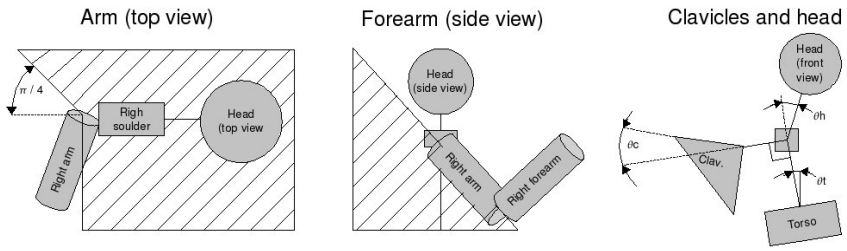


Figure 4: Articulations constraints. Arm and forearm: dashed lines show limb forbidden areas. The angular constraints are $|\theta_c| \leq 15^\circ$ for clavicles and $|\theta_h| \leq 25^\circ$ for head.

$$P(X_{0:T}|Y_{0:T}) = \prod_{t=0}^{T} \Phi(X_t, Y_t)\Psi(X_t) \prod_{t=1}^{T} T(X_t, X_{t-1}) \ , \tag{1}$$

with:

- $\Phi(X_t, Y_t) = \prod_{\mu=1}^{M} \phi^{\mu}(X_t^{\mu}, Y_t^{\mu})$,
- $\Psi(X_t) = \prod_{(\mu,\nu)\in\Gamma} \psi^{\mu\nu}(X_t^{\mu}, X_t^{\nu})$, where Γ is the set of links,
- $T(X_t, X_{t-1}) = \prod_{\mu=1}^{M} T^{\mu}(X_t^{\mu}, X_{t-1}^{\mu})$.

The marginal probabilities of the limbs' state are obtained using the belief propagation algorithm on a factor graph [3]. As the graph includes cycles, the obtained marginal is an approximation of the true one. This approximation further depends on the messages update order. To simplify the algorithm, the messages are propagated to all nodes within the current frame for a fixed number of iterations (10 in our case) and then propagated only once from a frame to the following one. Therefore, the estimation of a marginal at any time t does not depend on the observations after time t, and the estimation of the marginals can be computed recursively.

The messages are represented by sets of weighted samples. From one frame to the next, they are calculated using a particle filter scheme consisting in a re-sampling step followed by a prediction step based on the time coherence factors [4]. The loopy belief propagation algorithm is then reduced, for the current frame, to a loopy propagation algorithm for discrete state spaces, the space state for each limb being restricted to its samples. Moreover the marginal probability is then simply represented as a weighted sum of the same samples. In this manner, a full recursive estimation is obtained. The algorithm is equivalent to a set of interacting particle filters, where the sample weights are re-evaluated at each frame through belief propagation to take into account the links between limbs. This algorithm is relatively fast because for a frame t, as opposed to [11], the image based compatibility factors $\phi^{\mu}(X_t^{\mu}, Y_t^{\mu})$ have to be evaluated only once for each sample, and the link interaction factors only once for each pair of samples for all connected limbs.

3 Application to monocular upper body tracking

The model is applied to articulated upper-body tracking using monocular colour images from a webcam. Head and hands are tracked using image colour information and grey levels are used to compute cues: background subtraction, motion energy and orientation contour map (figure 1).

3.1 Initialisation

An accurate face detector [6] is used to detect the face in the colour image. Once detected, a starting pose corresponding to the arms along the body with the torso vertical and facing the camera is supposed. The tracker can easily recover the real pose as long as it is not too far from this hypothesis. The detected face is also used to initialise a face colour histogram.

3.2 Body model and link interaction factors

Figure 2 shows the body model. 3D limbs are represented by a sphere for the head and cylinders for arms and forearms. Hands, clavicles and torso are represented by 2D patches using respectively circles, triangles, and a rectangle. Limbs are discretized using a grid of regularly distributed points around them. A Gaussian of the distance between two link points is used to compute the link interaction factors (see figure 2 for distances Dn, Ds, De and Dw). This Gaussian is zero centred for the shoulder-arm and arm-forearm joints, and on a reference distance for the head-neck and forearm-hand joints. Other constraints are added giving zero factor for angles θh and θc above a fixed threshold (figure 4). Three additional links are defined, which simply give a zero probability to solutions where hands and head intersect (non collision constraints).

3.3 Time coherence factor

The time coherence factors $T^{\mu}(X_t^{\mu}, X_{t-1}^{\mu})$ are simple Gaussian, independent for each parameter, centred on the value in the previous frame. For hands, which can move fast and rapidly change speed, the time coherence factors is a mixture of two similar Gaussian, one centred on the previous parameter and the other centred on the prediction of the current parameter using previous hand speed. The standard deviation is chosen to be 10 cm for hands positions, and 5 cm for other limb positions. For angles, the standard deviation is set to $\pi/8$.

4 Image features

The image compatibility factors $\phi^{\mu}(X_t^{\mu}, Y_t^{\mu})$ are computed from scores S_f^{μ} representing the compatibility between a limb hypothesis μ and cue f extracted from the image. Contrary to stereo [3], monocular images needs more cues to reach a sufficient level of robustness. Thus, multicues image based compatibility terms are fused to provide an overall score: $S^{\mu} = \prod_f S_f^{\mu}$. To avoid taking into account background distractors, a robust background subtraction [9] is used.

Figure 5: Finding the torso. The bottom grid points (black pixels) representing the pelvis moves horizontally in order to maximise the correspondence between the points and the positive background subtraction pixels (white pixels). The maximum energy is reached when the grid is centred on the bottom positive background subtraction zone. The top of the torso is located at equal distance between the two clavicles.

4.1 Face and hands tracking

Considering the head position detected during initialisation step (§ 3.1), a colour model is provided by computing a normalised colour histogram of the head. The pixels p corresponding to the projected points belonging to the head or the hands are compared with this model by computing the colour score:

$$S_c^{\mu} = \sum_p H(p) \tag{2}$$

The function $H(p)$ returns the histogram bin value corresponding to the pixel p colour.

4.2 Torso tracking

The torso is hard to detect because of clothes deformations or occlusions produced when a person moves. Suposing that the pelvis is located at the bottom of images, its position can be found using a rectangular grid of weighted points p interacting with a background subtraction to slide on the bottom of the image (figure 5). The torso score is:

$$S^t = \sum_{p \in t} W(p) Bg(p) \tag{3}$$

Where $W(p)$ is the weight of p corresponding to the Gaussian distance between p and the grid center. $Bg(p)$ returns the probability that pixel p belongs to the foreground according to a background subtraction [9]. The upper torso point corresponds to the neck located at half distance of the two clavicles.

4.3 Arms, forearms and clavicles tracking

Arms tends to move rapidly and are subject to many partial occlusions. Thus, to reach a sufficient level of robustness, a fusion of a contour based cue and motion energy is implemented. An accurate contour based score can be estimated by not only considering the contours magnitude but also their orientations. Given $M(\|\overrightarrow{p}\|) = \frac{1}{\lambda}\|\overrightarrow{p}\| tanh(\frac{\lambda}{\|\overrightarrow{p}\|})$, a function that penalise low and high magnitude contour points $\|\overrightarrow{p}\|$ with λ a tuning parameter, a score S_{or}^{μ} for a limb hypothesis μ is computed by considering the Gaussian

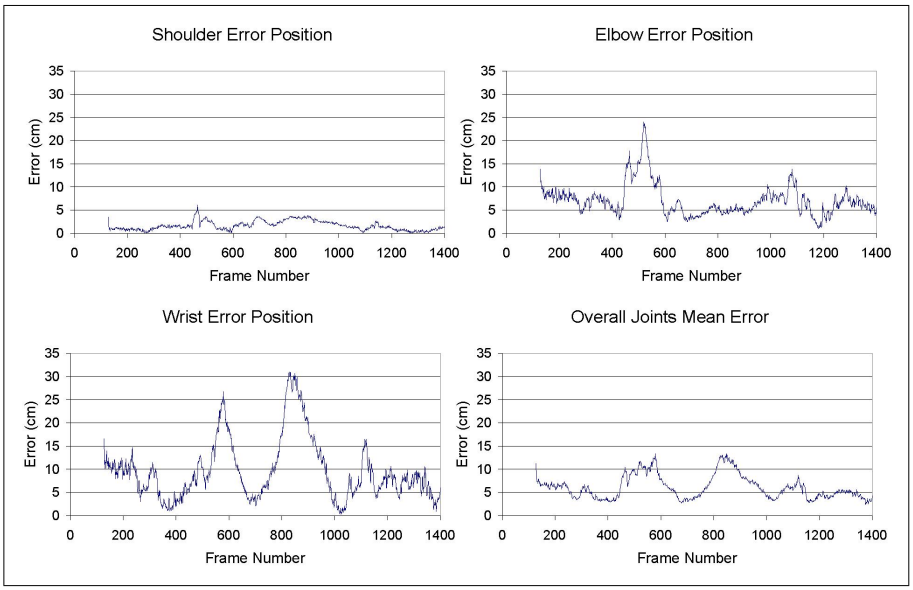


Figure 6: Quantitative results. For each joint, the error corresponds to the distance between estimated and true joint positions. As [12, 14], the mean error made on estimating the three joints is computed to provide the overall joint mean error.

difference $G_\theta(.)$ between the limb orientation θ_{limb} and each pixel contour orientation θ_p that corresponds to projected limb points p onto the image plane:

$$S_{or}^{\mu} = \sum_{p \in \mu} M(\|\overrightarrow{p}\|) G_\theta[\theta_{limb} - \theta_p] \tag{4}$$

The motion energy score is computed considering the Gaussian distance $G(d(p))$ between each projected limb point p and the nearest pixel where a motion has been detected: $S_m^{\mu} = 1 + \sum_{(p \in \mu)} G(d(p))$. Motion detection is provided by adjacent frame difference. This formula ensures that the motion score is at least 1 even if no motion is detected. Only the contour score is used for clavicles because they are strongly constrained by head position during belief propagation.

5 Experimental results

The system was tested on sequences grabbed with a standard webcam. Quantitative results were obtained comparing the estimated pose with a ground truth provided by a magnetic motion sensor. The true joint positions are measured for the right arm joints (shoulder, elbow and hand). The test sequence includes full 3D movements with limb occlusions and cluttered background (figure 9). Instead of only computing the overall limb mean error [12, 14], our results are complemented by the estimation error for each limb (figure 6). Qualitative results are shown on figure 7 where various user on different backgrounds and clothes are successfully tested.

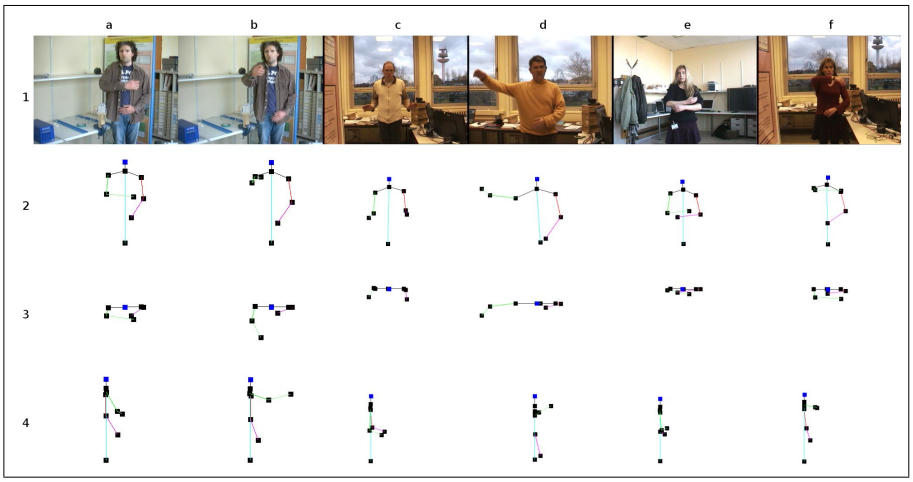


Figure 7: Monocular 3D tracking. Challenging poses are shown including occlusions, cluttered background and unconstrained environment (lighting and clothes).

Error (cm)	Shoulder	Elbow	Wrist	Overall Mean Error
Mean	1.7	7.1	9.7	6.1
Max	6.1	24.1	31.0	13.4
Std. Dev.	1.0	3.5	6.6	2.6
Average Speed ($cm.s^{-1}$)	2.83	4.28	8.5	

Table 1: Mean, maximum and standard deviation of the estimated position error for shoulder, elbow and wrist. Overall mean error is the mean error made on estimating the pose of theses three joints. Average speed is computed for the whole test sequence on each joint.

In monocular tracking, significant errors are usually made on depth estimation. It is the case in the test sequence around frame 500 owing to a wrong estimated elbow position that constrains the wrist in an exaggerated forward position. A similar problem occurs around frame 850 where forearm bends perpendicularly to the image plane and wrist depth is wrongly estimated by our algorithm (figure 8). Anyway, the maximal estimated pose error stays below 31 cm and below 15 cm considering the measure protocol used in [12, 14] (table 1). The comparison with other tracking algorithms is a difficult task owing to the disparity between used test sequences. However, the obtained results outperform or are as accurate than those computed with existing algorithms [12, 14].

6 Conclusion

We have presented an algorithm for monocular upper body tracking performing at 6 *fps* using a standard webcam with unconstrained environments (lighting and clothes). The used cues based on contours provide sufficient robustness to succeed on unconstrained environments. Belief propagation provides a judicious solution in order to reduce the

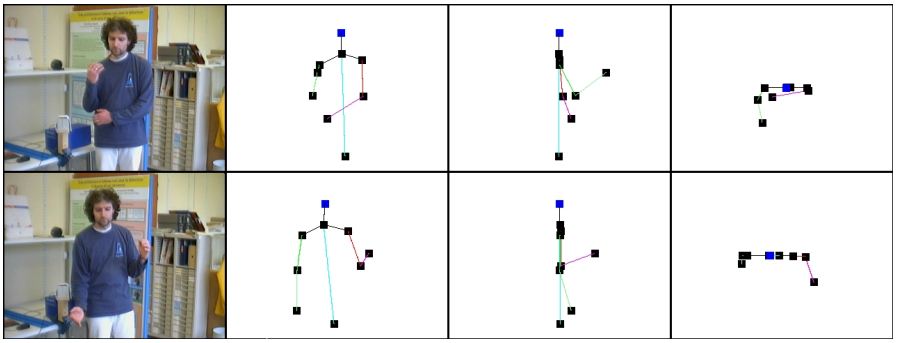


Figure 8: Examples of wrong depth estimation on frames 581 (first row) and 850 (second row). In both cases, right forearm is not bended sufficiently involving errors larger than 25 cm on wrist pose estimation.

space dimension of the generated hypothesises making particle filtering framework suitable. Articulation constraints are easily integrated into factors computation to provide consistent resulting poses. Future work will include a learning based image compatibility term to handle occlusions and more accurate depth estimation.

References

[1] Ankur Agarwal and Bill Triggs. A local basis representation for estimating human pose from cluttered images. In *ACCV (1)*, pages 50–59, 2006.

[2] Ankur Agarwal and Bill Triggs. Recovering 3d human pose from monocular images. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 28(1), jan 2006.

[3] Olivier Bernier and Pascal Cheung-Mon-Chang. Real-time 3d articulated pose tracking using particle filtering and belief propagation on factor graphs. In *British Machine Vision Conference*, volume 01, pages 005–008, 2006.

[4] Andrew Blake and Michael Isard. The condensation algorithm - conditional density propagation and applications to visual tracking. In *NIPS*, pages 361–367, 1996.

[5] David Demirdjian, T. Ko, and Trevor Darrell. Constraining human body tracking. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 1071. IEEE Computer Society, 2003.

[6] Raphaël Féraud, Olivier Bernier, Jean Emmanuel Viallet, and Michel Collobert. A fast and accurate face detector based on neural networks. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pages 23(1):42–53, 2001.

[7] Jiang Gao and Jianbo Shi. Multiple frame motion inference using belief propagation. In *FGR*, pages 875–882, 2004.

[8] Kschischang, Frey, and Loeliger. Factor graphs and the sum-product algorithm. *IEEETIT: IEEE Transactions on Information Theory*, 47, 2001.

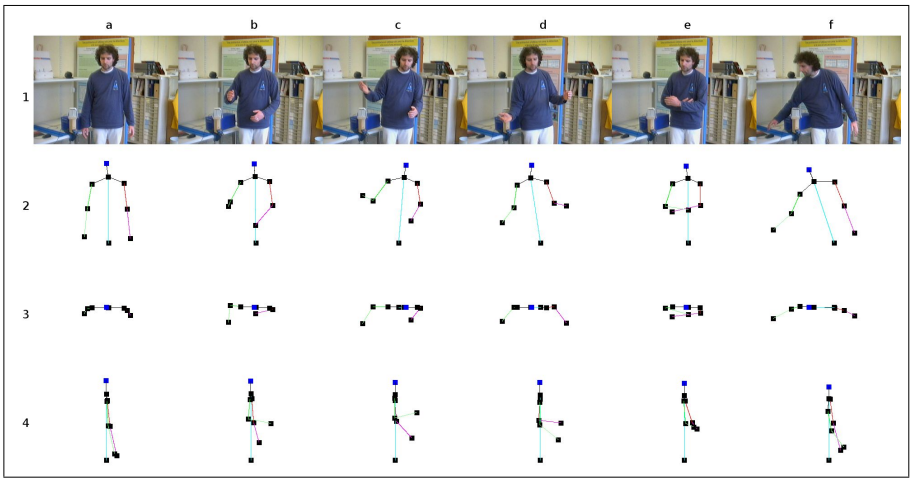


Figure 9: Test sequence from which the ground truth were measured. From column (a) to (f): frames 246 (initialisation), 409, 709, 813, 1184 and 1437. The face, top and right side estimated poses are shown on row 2, 3 and 4.

[9] Philippe Noriega and Olivier Bernier. Real time illumination invariant background subtraction using local kernel histograms. In *Proceedings of the British Machine Vision Conference*, pages 979–988, 2006.

[10] Gregory Shakhnarovich, Paul Viola, and Trevor Darrell. Fast pose estimation with parameter-sensitive hashing. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 750. IEEE Computer Society, 2003.

[11] Leonid Sigal, Sidharth Bhatia, Stefan Roth, Michael J. Black, and Michael Isard. Tracking loose-limbed people. In *CVPR (1)*, pages 421–428, 2004.

[12] Leonid Sigal and Michael J. Black. Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2041–2048, Washington, DC, USA, 2006. IEEE Computer Society.

[13] Cristian Sminchisescu and Alexandru Telea. Human pose estimation from silhouettes - a consistent approach using distance level sets. In *WSCG*, pages 413–420, 2002.

[14] Leonid Taycher, David Demirdjian, Trevor Darrell, and Gregory Shakhnarovich. Conditional random people: Tracking humans with crfs and grid filters. In *CVPR (1)*, pages 222–229, 2006.

[15] Raquel Urtasun, David J. Fleet, and Pascal Fua. 3d people tracking with gaussian process dynamical models. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 238–245, Washington, DC, USA, 2006. IEEE Computer Society.

Robust Multi-View Change Detection

Alessandro Lanza Luigi Di Stefano
University of Bologna – DEIS – ARCES
40136 Bologna, Italy
{alanza,ldistefano}@deis.unibo.it

Jérôme Berclaz François Fleuret Pascal Fua
EPFL – CVLAB
CH – 1015 Lausanne, Switzerland
{jerome.berclaz,francois.fleuret,pascal.fua}@epfl.ch

Abstract

We present a multi-view change detection approach aimed at being robust with respect to common "disturbance factors" yielding image changes in real-world applications. Disturbance factors causing "slow" or "fast-and-global" image variations, such as light changes and dynamic adjustments of camera parameters (e.g. auto-exposure and auto-gain control), are dealt with by a proper single-view change detector run independently on each view. The computed change masks are then fused into a "synergy mask" defined into a common virtual top-view, so as to detect and filter-out "fast-and-local" image changes due to physical points lying on the ground surface (e.g. shadows cast by moving objects and light spots hitting the ground surface).

1 Introduction

Detecting changes in video sequences plays a crucial role in many computer vision applications since the performance of higher-level processing modules, such as objects tracking and classification, often relies on the accuracy of the computed change masks. In the space of all the possible image changes a *good* change detector should be able to discriminate between "semantic" (i.e. due to variations of the scene geometry) and "appearance" (i.e. due to other causes, that we call "disturbance factors") changes. In particular, a change detection algorithm should be robust with respect to disturbance factors arising both in the imaged scene (e.g illumination changes) and in the imaging device (e.g. noise, dynamic adjustments of device parameters such as auto-exposure and auto-gain control).

Most of the single-view change detectors proposed in literature (e.g. [3], [10]) can deal effectively with camera noise and "slow" scene appearance changes (e.g. scene illumination changes due to time of the day). To this purpose, a temporally adaptive per-pixel statistical modelling of the scene background appearance is exploited. To avoid the inclusion of foreground objects in the background appearance model, the model adaptation rate must be chosen accurately, depending on the foreground objects foreseen velocity. In particular, the lower the foreground objects foreseen velocity, the lower the background model adaptation rate. Hence, in general only quite slow appearance changes can be dealt with by these algorithms. Some approaches have been proposed (e.g. [2],[7],[9],[11])

which can deal effectively also with "fast-and-global" scene appearance changes, that is fast changes modifying pixel intensities by a unique mapping function. Examples of such changes are those due to fast-and-global scene illumination changes (e.g. light switches, a cloud passing by the sun) and to dynamic adjustments of camera parameters (e.g. auto-exposure and auto-gain control). "Fast-and-local" scene appearance changes (e.g. shadows cast by moving objects, light spots hitting a nearly lambertian surface) are a hard-to-solve problem for single-view approaches.

Multi-view change detection can exploit more information and therefore deal more effectively with disturbance factors. As regards the way information is exploited, we define:

c.1) *temporal consistency* constraint: given a view-point v, the processed frames are images of the same scene taken at different times;

c.2) *spatial coherence* constraint: given a time t, the processed frames are images of the same scene taken from different view-points;

By applying only the spatial coherence constraint the basic multi-view change detection approach is carried out. In practice, at each time t the output is computed by comparing all the simultaneous images captured from the different view-points. However, all the available information can be exploited by applying both the constraints. This is in theory the most effective approach. We present a multi-view change detection algorithm of this type. In particular, we apply the temporal consistency constraint as a first processing step by carrying out single-view change detection on each original view. Then, the spatial coherence constraint is applied by "fusing" the single-view change masks into a virtual top-view. Such an approach allows for filtering-out the appearance changes due to the major disturbance factors, including sudden-and-local illumination changes.

The paper is organized as follows. In section 2 the state-of-the-art in multi-view change detection is outlined. The proposed algorithm is presented in section 3. Experimental results are discussed in section 4, conclusions are drawn in section 5.

2 Related Work

In [5] a "lighting independent" multi-view change detection algorithm is presented. Stationarity of the capturing devices as well as of the scene background surface geometry is assumed, so that the geometric transformations warping one of the views, called "primary" view, into all the other "auxiliary" views can be computed off-line. On-line, just the change mask in the primary view is computed. Moreover, only the spatial coherence constraint is applied. In practice, at each time, the colour of every pixel in the primary view is compared with the colour of corresponding pixels in the auxiliary views, using the geometric transformations. If colour is similar, according to a simple metric consisting in the absolute value of the Euclidean distance, the pixel in the primary view is marked as background; otherwise, it is marked as foreground. This approach inherently suffers from both false and missed detections. False detections, called "occlusion shadows", occur when a background pixel in the primary view is occluded by a foreground object in the auxiliary view. Missed detections occur when an evenly coloured foreground object occludes a pair of corresponding pixels, for colour being very similar. The authors propose to filter-out false detections by using more than two views (at least two auxiliary

views) and ANDing the binary masks attained by comparing the primary view to each of the auxiliary views. However, they do not discuss how to deal with missed detections.

The work in [8] is aimed at improving the approach proposed in [5]. As in [5], the change mask in the primary view is computed by applying only the spatial coherence constraint. However, the following improvements are proposed:

a) a slightly more complex and effective metric (i.e. a normalized colour difference averaged on a $n \times n$ neighborhood of pixels) is used to measure colour similarity between corresponding pixels in different views;

b) the false detections problem is addressed from a sensor planning perspective. In particular, it is shown how occlusion shadows can be removed by using just two views, provided that a suitable configuration of the capturing devices is adopted;

c) the missed detections problem is tackled as well. The particular sensors configuration adopted to filter-out occlusion shadows yields missed detections localized only at the lower portion of each detected foreground blob. This is exploited to fill-in possible missed detections by means of a quite complex heuristic procedure.

Both [5] and [8] rely on the assumption that a patch of the scene background surface yields a very similar colour into simultaneous images taken from different view-points. If this is true, invariance to temporal changes of the radiance emitted by the scene background surface (i.e. to slow or fast and global or local scene illumination changes) is achieved, since such changes will affect simultaneous views identically. However, in practice this assumption may not be satisfied. In fact, dynamic adjustments of the camera parameters (e.g. auto-gain and auto-exposure control) may occur in the different views at different times and by a different intensity mapping function. These adjustments cannot be handled inherently by either [5] or [8]. In turn, [5] recommends explicitly to disable the auto-gain mechanism of the capturing devices. However, disabling these dynamic adjustment mechanisms is a strong limitation in many practical applications, especially as regards outdoor installations.

The most related work to our approach is presented in [6]. It is focused on tracking but relies on multi-view change detection as the first processing step. People moving on a ground plane are tracked by their ground locations, that is feet. At each processing time feet are detected by a multi-view change detection approach, that we call here "change maps fusion": the ground plane homographies warping a reference view into each of the other views are inferred off-line. On-line, single-view change detection is carried out independently on each view to compute a change probability map. To this purpose, a well-known background subtraction algorithm based on statistical temporally adaptive background modelling by mixture of gaussians is deployed ([10]). Hence, the computed change probability maps are warped in the reference view by using the inferred homographies and then multiplied together, thus attaining a "synergy map". It is easy to understand how this map gives, for each pixel in the reference view, the probability to be the image of a ground plane patch for which the emitted radiance is changed (with respect to the current appearance background model and according to the chosen single-view change detection algorithm). Finally, the synergy map is thresholded. By this procedure, the authors assume to detect only the ground plane locations of people, that is their feet. Hence, feet are tracked in the reference view by a spatio-temporal clustering approach (graph cuts). However, the proposed use of the change maps fusion approach

will inherently detect as foreground not just feet but also other possible ground plane appearance changes, such as shadows cast by moving objects on the ground plane or light spots hitting the ground plane. In fact, such changes are not filtered-out by the single-view change detection approach in [10].

3 The proposed algorithm

We assume stationarity of the capturing devices as well as of the scene background surface geometry, so that geometric registration of background over different views can be computed off-line. Moreover, we take into consideration a planar background, hereinafter called "ground plane". Hence, for each original view v, we infer off-line the homography $H^v : \mathbb{R}^2 \ni \boldsymbol{p}^v \mapsto \boldsymbol{p}^T \in \mathbb{R}^2$ warping each pixel $\boldsymbol{p}^v$ imaging a ground plane patch in the original view into the pixel $\boldsymbol{p}^T$ imaging the same patch in a common virtual top-view T. By considering a set of $N > 4$ original view $\leftrightarrow$ top-view points correspondences, the homographies are inferred by least squares regression. A data normalization procedure is adopted to make the necessary matrix calculations less prone to numerical errors ([4]).

As far as on-line processing is concerned (Figure 1), at each time t first the temporal consistency constraint is applied by carrying out single-view change detection independently on each original view ([2],[7]), thus computing a set of V binary change masks C_t^v, one for each original view $v = 1, \ldots, V$ (Figures 1(d-f)). The spatial coherence constraint is then applied by projecting all the change masks[1] into the virtual top-view, thus attaining a set of V top-view change masks $C_t^{v,T}$ (Figures 1(g-i)):

$$C_t^{v,T} = H^v(C_t^v) \tag{1}$$

Then, a common top-view change mask C_t^T is obtained by computing the intersection of all the top-view change masks (Figures 1(j)):

$$C_t^T = \bigcap_{v=1}^{V} C_t^{v,T} \tag{2}$$

The procedure outlined so far is substantially equivalent to the change maps fusion approach presented in [6]. The only difference is that we carry out change maps binarization directly as the final step of the temporal consistency constraint application. On the other hand, in [6] binarization is carried out in the virtual top-view after the spatial coherence constraint has been applied as well. We call "change masks fusion" this slightly different approach and "synergy mask" the binary mask of Equation 2. However, we deploy the *synergy* information within the top-view in a "dual" manner with respect to [6]. In fact, the synergy mask contains the pixels characterized by a high probability to be the image of a ground plane patch for which the emitted radiance is changed. These pixels correspond to people feet as well as to possible ground plane appearance changes, such as those due to shadows cast by people or to light spots hitting the ground plane. Therefore, instead of using the synergy mask to detect foreground objects ground locations (people feet), we use it to filter-out ground plane appearance changes, like shadows or light spots. In particular, instead of considering the synergy mask as the final output of the multi-view change detection, we back-project the synergy mask into all the original views, thus obtaining a

[1] actually, just the change masks portion inside the ground plane limits are projected

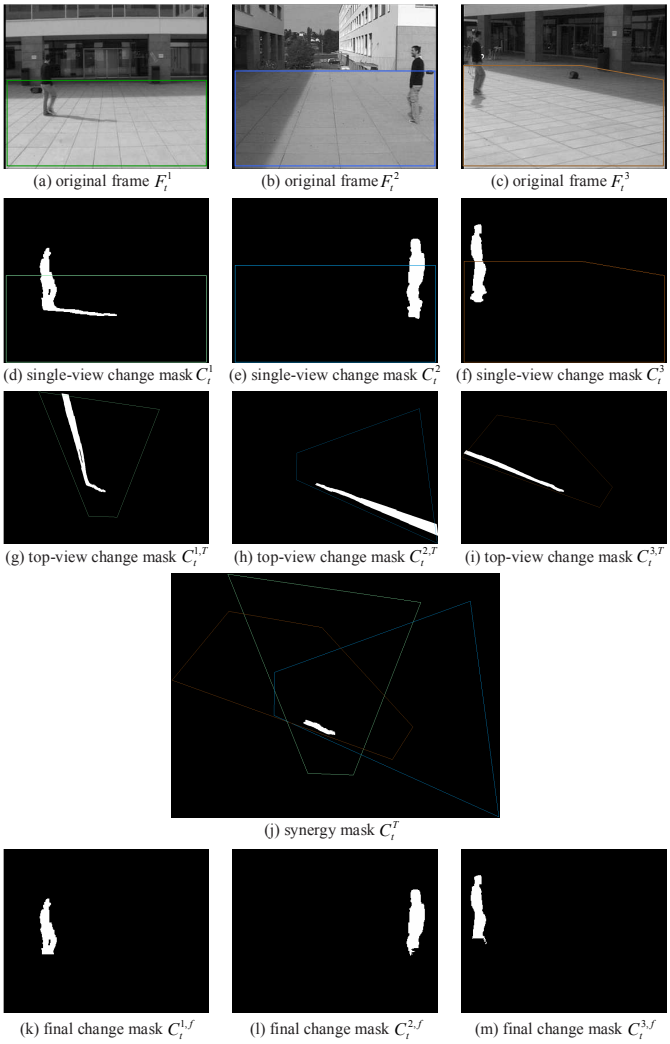


Figure 1: On-line main processing steps of the proposed multi-view approach.

set of V original view synergy masks $C_t^{T,v}$:

$$C_t^{T,v} = (H^v)^{-1}\left(C_t^T\right) \tag{3}$$

Then, for each view v we filter-out from the original view change mask C_t^v the foreground pixels belonging to the original view synergy mask $C_t^{T,v}$, thus attaining a set of V final change masks $C_t^{v,f}$ (Figures 1(k-m)):

$$C_t^{v,f}(\boldsymbol{p}^v) = \begin{cases} 0 & \text{if} \quad C_t^{T,v}(\boldsymbol{p}^v) = 1 \\ C_t^v(\boldsymbol{p}^v) & \text{otherwise} \end{cases} \tag{4}$$

Hence, another difference with respect to [6] is that we compute a set of V change masks, one for each original view, instead of a single change mask in the virtual top-view. Moreover, the change masks will include most of a person's body (ideally, the entire body but the feet). Unlike [5] and [8], our approach handles dynamic adjustments of camera parameters provided that a proper change detection algorithm (i.e. [2],[7]) is run on each original view. It is worth pointing out that algorithms such as [2] and [7] can also deal very effectively with sudden and global light changes.

The proposed approach is "general-purpose", in the sense that all the scene appearance changes detected by the employed single-view change detection algorithm which satisfy the spatial coherence constraint (i.e. which arise "near" the ground plane in a 3-dimensional sense) are filtered-out. In fact, no selectivity criterion is used in the removing rule of expression 4. In practice, just a geometrical constraint is applied, without considering any photometric information. On one hand this approach is general-purpose, but on the other hand a missed detections problem may arise due to the following two causes:

a) part of the foreground objects ground locations, especially people feet, may be removed together with the actual false changes (e.g. shadows) from the final change masks (Figure 1(k)). This is an inherent and easy to understand problem of the proposed approach, since ground locations of foreground objects yield appearance changes lying "near" the ground plane (i.e. they satisfy the spatial coherence constraint);

b) some "off-ground" portions of the foreground objects may be removed as well. This may occur for the original views in which the ground plane appearance changes are covered by foreground objects (Figure 1(l)). This is an inherent problem as well. In general, the higher the number of foreground objects present in the scene, the higher the probability of this problem to occur.

To face these two inherent problems we propose a less "general-purpose" removing rule, that we call "shadows-focused" removing rule. In fact, by this new rule we try to achieve a selective removal of just the ground plane appearance changes due to shadows. To this purpose, we exploit simple, well-known and commonly used photometric properties characterizing scene surfaces covered by shadows. The basic idea is that the measured intensity of a pixel imaging a scene background surface patch decreases according to a limited darkening factor d when covered by a cast shadow. Hence, the selective "shadows-focused" removing rule is the following:

$$C_t^{v,f}(\boldsymbol{p}) = \begin{cases} 0 & \text{if } \left(C_t^{T,v}(\boldsymbol{p}^v) = 1\right) \wedge \left(d_{low} < \dfrac{F_t^v(\boldsymbol{p}^v)}{\hat{B}_t^v(\boldsymbol{p}^v)} < 1\right) \\ C_t^v(\boldsymbol{p}^v) & \text{otherwise} \end{cases} \tag{5}$$

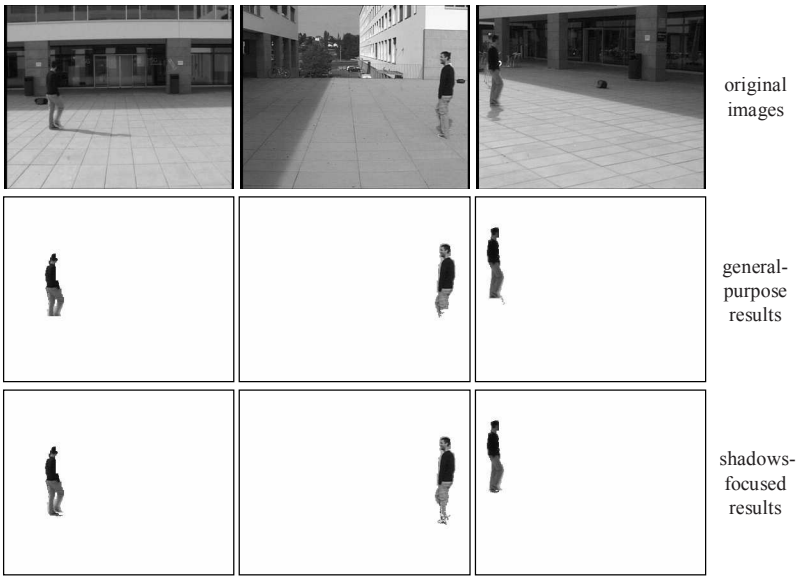
where d_{low} is the lower darkening factor assumed for shadows effect and F_t^v, $\hat{B}_t^v$ are, respectively, the current frame and the current background model used by the single-view change detection algorithm in the original view v. In practice, for each view v the final change mask $C_t^{v,f}$ is not computed by filtering-out blindly all the foreground pixels of the original view synergy mask $C_t^{T,v}$ from the original view change mask C_t^v. Instead, just the foreground pixels satisfying the shadows photometric constraint are removed.

4 Experimental Results

Experiments have been carried out by running the proposed general-purpose and shadows-focused multi-view change detection approaches on several test video sequences. All the sequences have been captured by the same multi-view outdoor installation, consisting of three synchronized capturing devices imaging a common scene from very different view-points. Within the imaged scene, people walk and cast shadows on a planar ground. Here we present the change detection results for four different processing times (i.e. for four different triples of simultaneous frames) of a test sequence. In particular, the change masks computed by the general-purpose (blind removing rule of Equation 4) and by the shadows-focused (selective removing rule of Equation 5) approaches are directly compared in Figures 2-3. In particular, a value $d_{low} = 0.5$ is used in the shadows-focused removing rule. Shadows cast by moving people on the ground plane are removed effectively by both the approaches. In fact, since shadows seen in each view lie on the ground plane their entire shapes will be projected into the synergy mask and hence detected. This works well for long as well as short shadows. However, the general-purpose approach suffers from a missed detections problem, as expected. On one hand, in each view people feet may be partially removed, independently from the reciprocal position of people and cast shadows. In fact, feet yield a local change of the radiance emitted by the ground plane. As an example, the change masks on the left and on the right of the centre row of Figures 2(a,b) show how feet can be partially removed also in the very favourable situation of a single person moving in the scene without covering its cast shadow. On the other hand, "off-ground" portions of people's body may be removed as well when cast shadows are covered by people. This is the case of Figure 2(a), top row, in the middle, where the person covers almost completely its cast shadow. As a consequence, the lower portion of the person's body, that is the portion covering the cast shadow, is detected as unchanged, as shown in Figure 2(a), centre row, in the middle. In general, the higher the number of persons present in the scene, the higher the probability of this problem to occur, as shown in Figures 3(a,b), centre row. As for the considered test sequences, the missed detections problem is solved quite effectively by the shadows-focused approach, as regards both the feet and the covered shadows problems (Figures 2-3(a,b), bottom row). However, it is worth noticing that in general the persons' body appearance impacts the actual effectiveness of the shadows-focused approach in dealing with the missed detections problem. Finally, we point out that a shadow removal approach based only on the application of the photometric constraint in Equation 5 would be prone to the detection of false shadows not lying on the ground plane.

5 Conclusions

We have presented a multi-view change detection approach aimed at being robust to the major disturbance factors acting in real-world applications. On one hand, camera noise and disturbance factors yielding slow or global background appearance changes are dealt with by single-view change detection carried out independently on each original view. On the other hand, fast-and-local appearance changes are filtered-out by fusing the single-view change masks into a common virtual top-view and then back-projecting the attained synergy mask into the original views. However, sudden changes due to specular reflections can not be dealt with by the proposed algorithm for the ground plane constraint does



(a) frame 76

(b) frame 133

Figure 2: Change masks computed by the proposed general-purpose (centre row of (a) and (b)) and shadows-focused (bottom row of (a) and (b)) change detection approaches for frames 76 (top row of (a)) and 133 (top row of (b)).

(a) frame 333

(b) frame 355

Figure 3: Change masks computed by the proposed general-purpose (centre row of (a) and (b)) and shadows-focused (bottom row of (a) and (b)) change detection approaches for frames 333 (top row of (a)) and 355 (top row of (b)).

not hold in this case. Since a missed detections problem may arise due to causes which are inherent to the presented approach, a less general-purpose version of the algorithm has been proposed as well, focused on shadows removal. Since the appearance changes occurring in the available multi-view test sequences are all due to shadows cast by moving people on the ground plane, the shadows-focused approach yields better results than the general-purpose approach, as shown by experiments. Unlike other state-of-the-art multi-view change detection algorithms, which compute a single change mask in a reference ([5],[8]) or a virtual ([6]) view, the output of our approach is a set of different change masks, one for each original view. This output is suitable to be fed to a multi-view tracking algorithm such as ([1]).

References

[1] J. Berclaz, F. Fleuret, and P. Fua. Robust people tracking with global trajectory optimization. In *Proc. CVPR'06*, volume 1, pages 744–750, June 2006.

[2] A. Bevilacqua, L. Di Stefano, and A. Lanza. Coarse-to-fine strategy for robust and efficient change detectors. In *Proc. AVSS'05*, pages 87–92, September 2005.

[3] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proceedings of the IEEE*, 90(7):1151–1163, July 2002.

[4] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[5] Y. A. Ivanov, A. F. Bobick, and J. Liu. Fast lighting independent background subtraction. *International Journal of Computer Vision*, 37(2):199–207, June 2000.

[6] S. M. Khan and M. Shah. A multiview approach to tracking people in crowded scenes using a planar homography constraint. In *Proc. ECCV'06*, volume 4, pages 133–146, May 2006.

[7] A. Lanza and L. Di Stefano. Detecting changes in grey level sequences by ML isotonic regression. In *Proc. AVSS'06*, pages 4–4, November 2006.

[8] S. N. Lim, A. Mittal, L. S. Davis, and N. Paragios. Fast illumination-invariant background subtraction using two-views: Error analysis, sensor placement and applications. In *Proc. CVPR'05*, volume 1, pages 1071–1078, June 2005.

[9] N. Ohta. A statistical approach to background subtraction for surveillance systems. In *Proc. ICCV'01*, volume 2, pages 481–486, July 2001.

[10] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Proc. CVPR'99*, volume 2, pages 246–252, June 1999.

[11] B. Xie, V. Ramesh, and T. Boult. Sudden illumination change detection using order consistency. *Image and Vision Computing*, 22(2):117–125, February 2004.

Higher-order Autoregressive Models for Dynamic Textures

Midori Hyndman, Allan Jepson, David Fleet
Department of Computer Science
University of Toronto
mhyndman,jepson,fleet@cs.toronto.edu

Abstract

Dynamic textured sequences are characterized by the interactions between many particles or objects in the scene. Based on earlier work the images of the sequence are interpreted as the output of a linear autoregressive process driven by white Gaussian noise. We extend earlier work by increasing the amount temporal information included when learning the motion in the scene, allowing the models to capture complex motion patterns which extend over multiple frames, thereby increasing the perceptual accuracy of the synthesized results. To overcome problems of dynamic model stability, we apply Burg's Maximum Entropy Spectral Analysis technique for parameter estimation, which is found to be reliably stable on smaller samples of training data, even with higher-order dynamics.

1 Introduction

A dynamic texture is an image sequence characterized by the interactions between many particles or objects in the scene. Examples of dynamic textures include, flames flickering, leaves blowing, and crowds observed from a distance. For such scenes, learning the motion by segmenting and tracking the trajectory of each component is computationally intensive; a holistic representation of the scene and the motion is motivated.

One well-known approach is to infer linear, autoregressive models of dynamic textures. The frames of the image sequence are interpreted as the output of stochastic process driven by white Gaussian noise. The appearance of the scene is described by a subspace model and the dynamics of the scene are captured within this subspace by a generative model that determines the hidden state of the system. Previous work using autoregressive models for dynamic texture synthesis, [6] in particular, used a first-order dynamical model. Incorporating only information from the preceeding state prevents the capture of oscillations and other motions that rely on higher-order temporal dependencies in the image sequence. Also, with first-order models the perceptual quality of the synthesized scene deteriorates within a short interval.

In this paper, we propose the use of higher-order autoregressive dynamic texture models. We find that increasing the amount of temporal information when learning the interframe dependencies allows the model to capture complex patterns which extend over multiple frames, increasing the perceptual accuracy of the synthesized results.

When incorporating a higher-order dynamical model, issues of model stability arise. To overcome these issues we apply the Maximum Entropy Spectral Analysis (MESA)

technique for linear prediction [3]. This approach is common in control theory but, to our knowledge, not typically used in the field of computer vision and new to dynamic texture modeling. This estimation technique is more reliably stable and perceptually accurate on smaller samples of training data, even with a higher-order dynamical model, than when using the Yule-Walker equations.

2 Related Work

Texture analysis and synthesis was pioneered by Julesz [12] with the observation of the correlation between statistical and perceptual similarity of textured images. Since then, many image-based rendering techniques have emerged for synthesizing static and dynamic textured scenes.

Non-parametric methods synthesize images using probabilistic sampling of the observed data, either pixel by pixel [5, 9, 10, 27] or by copying patches [8, 15, 28]. In non-parametric dynamic texture synthesis, notable results have emerged using patch-based techniques, where image patches are interpreted as segments of the image sequence [14, 22]. The synthesized temporal textures generated with these methods tend to be perceptually realistic, however, the images are limited to samples of the original sequence. Moreover, because a model of the scene is not explicitly inferred, the synthesized results cannot be generalized and further processing, viz. classification [4], is limited.

Parametric methods for dynamic textures were introduced in [18]. Modeling dynamic textures as the output of a spatio-temporal autoregressive process was shown to be successful with certain classes of textures and motions [25], however, the framework could not model spatially non-stationary motion, such as rotation. In [6], these limitations are addressed by representing dynamic textures as the output of a first-order subspace process with a Gaussian driving distribution,

$$y_t = Cx_t \tag{1}$$

$$x_t = -Ax_{t-1} + Wv_t, \quad v_t \sim \mathcal{N}(0, I). \tag{2}$$

In their appearance model (1) each image y_t is considered an expansion of the state variable x_t which is defined in the principal component subspace. In their dynamic model (2) the current hidden state of the system is derived from a linear combination of the elements in the preceeding state, described by matrix A, and additive Gaussian noise with covariance WW^T is used to stochastically drive the process.

In [6], the dynamic model parameters are learned within the appearance model subspace. If the dynamic and appearance information is non-separable, this approach determines only an approximation to the optimal parameter estimates. To guarantee an optimal parameter estimate, the appearance and dynamic model parameters would be learned simultaneously. In [26] the dynamic model is computed in the original input space and the appearance model is constructed to retain a maximum amount of the information with respect to the dynamics of the system. In [23] an iterative approach is suggested where the results of [26] are used for initialization. Unfortunately these techniques are computationally infeasible on common workstations, given high-dimension input such as image data. Instead, we implement a closed-form solution to approximate the optimal parameter estimates, as in [6].

In contrast to the prevalent use of first-order dynamical models in earlier work, we advocate the use of higher-order models in the autoregressive process. We show that

higher-order models produce improved synthesized sequences with perceptual quality maintained over a longer time interval. The advantages of the autoregressive framework are preserved: separating the appearance and dynamical components enables classification [4], facilitates recognition applications [21] and provides a more manipulable model to explore video editing [7]. Moreover, incorporating influence from states lagged further in time captures the temporal dependencies that are capable of modeling oscillations. Using higher-order dynamics, however, introduces issues of model stability. We draw on a parameter estimation technique used in control theory to improve the stability of the resulting model, the Maximum Entropy Spectral Analysis technique [3].

3 Autoregressive Model

In this work a dynamic texture is modeled as the output of an autoregressive process consisting of an appearance model, which determines the state of the system, and a dynamic model, which captures how the states change over time:

$$y_t = Cx_t + u_t, \qquad u_t \sim \mathcal{N}(0,B), \tag{3}$$

$$x_t = -\sum_{i=1}^{\mu} F_{\mu,i}x_{t-i} + Wv_t, \quad v_t \sim \mathcal{N}(0,I). \tag{4}$$

At time t, each image y_t, in column vector form, is defined by the expansion of a hidden state variable, x_t. In the generative appearance model (3) the matrix C projects the subspace representation into the image space, and the zero-mean normally distributed additive noise captures the uncertainty with covariance B. The dynamic model (4) contains a deterministic component (i.e. a Markov-model described by $F = \{F_{\mu,1}, F_{\mu,2}, \ldots, F_{\mu,\mu}\}$) and a stochastic component (i.e. a Gaussian driving distribution with covariance WW^T). As in [6], we ignore the additive appearance noise u_t (i.e., take $u_t \equiv 0$) and capture all additive process noise within the driving distribution v_t.

We learn the parameters C, F, and W for the μ-order autoregressive model of an image sequence. Initializing the model with a set of μ consecutive image frames, one can generate novel image sequences which resemble the original data. The model is successful if the synthetic sequences are perceptually similar to the original sequence and, ideally, the model parameters are sufficiently generalizable to support recognition tasks [21].

3.1 Appearance Model

While the optimal estimator finds C, F, and W simultaneously, following [6], we use principal component analysis (PCA) to define the appearance model parameters C and we learn the dynamical model parameters F and W within the PCA subspace. To determine C, each image of the observed sequence is converted into column vector form, the mean image is subtracted, and the resulting vectors are concatenated to form Y_1^τ, a matrix of size $p \times \tau$ where p is the number of pixels per image times the number of colour channels, and τ is the number of images ($\tau < p$). Let $Y_1^\tau \equiv U\Sigma V^T$ be a singular value decomposition (SVD) where U is $p \times p$, Σ is $p \times \tau$, and V is $\tau \times \tau$. We choose $q \ll p$ and define $C \equiv \widehat{U}$ where $\widehat{U}$ is a matrix containing the first q principal directions found in the columns of U. Let $\widehat{V}$ be the first q columns of V and $\widehat{\Sigma}$ be a diagonal matrix of the q largest singular values from Σ. We define the subspace representation of Y_1^τ to be $X_1^\tau \equiv \widehat{\Sigma}\widehat{V}^T$. There are non-linear alternatives which, in future work, could be used within the appearance model; in particular, [20] is developed specifically for spatial textures.

3.2 Dynamic Model

The dynamic model comprises a deterministic linear model and a Gaussian driving distribution. The true dynamical process which generated the orignal sequence may contain both linear and non-linear components. Nonetheless, we assume that a linear autoregressive model is sufficient to describe the visual process. Information not captured within the linear component is modeled in the stochastic component of the dynamics.

The Yule-Walker equations can be used to solve for the coefficients of the dynamic model in the least squares sense, as in [6]. However, this approach assumes the stationarity of the training data sample statistics, an assumption which breaks down for short dynamic texture segments. As the order of the dynamic model increases and accuracy of the sample statistics deteriorate, the dynamic model determined with the Yule-Walker method is often unstable. In an unstable linear system, the predicted states tend towards infinity over time, resulting in perceptually unrealistic synthesized sequences.

The Maximum Entropy Spectral Analysis (MESA) technique was developed for single channel signals [3] and extended to handle multidimensional data [16, 24]. Although common in the control theory literature, to our knowledge this technique has not been applied to dynamic textures. When modeling dynamic textures, in practice only small portions of the sequences are available. Inaccurate models result when the higher-order sample statistics do not adequately reflect the structure in data. The main contribution of MESA is that by using a recursive approach the higher-order autocorrelations are never calculated directly from the sample data, despite the assumption of stationarity. An additional advantage of MESA, is that the stability of the resulting model is guaranteed [13]. Moreover, compared to using the Yule-Walker equations, we found that fewer training frames are necessary to obtain an accurate model [11].

MESA uses a recursive approach that depends on the coefficients of both forward and backward models,

$$x_t = -\sum_{i=1}^{\mu} F_{\mu,i} x_{t-i} + e_{\mu,t}, \tag{5}$$

$$x_t = -\sum_{i=1}^{\mu} B_{\mu,i} x_{t+i} + b_{\mu,t}. \tag{6}$$

where $e_{\mu,t}$ and $b_{\mu,t}$ are the forward and backward residuals. In (5), future states are predicted using the past states of the system, whereas in (6) past states are predicted using future data. The coefficients of an μ-order model are as follows,

$$\mathbf{F}_\mu = \begin{bmatrix} I & F_{\mu,1} & F_{\mu,2} & \dots & F_{\mu,\mu} \end{bmatrix}^T, \tag{7}$$

$$\mathbf{B}_\mu = \begin{bmatrix} B_{\mu,\mu} & \dots & B_{\mu,2} & B_{\mu,1} & I \end{bmatrix}^T, \tag{8}$$

where I is an identity matrix of size $q \times q$. These model coefficients have the following recursive relationship [3],

$$\mathbf{F}_\mu = \begin{bmatrix} \mathbf{F}_{\mu-1} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{B}_{\mu-1} \end{bmatrix} F_{\mu,\mu}, \tag{9}$$

$$\mathbf{B}_\mu = \begin{bmatrix} \mathbf{F}_{\mu-1} \\ 0 \end{bmatrix} B_{\mu,\mu} + \begin{bmatrix} 0 \\ \mathbf{B}_{\mu-1} \end{bmatrix}. \tag{10}$$

Matrices $F_{\mu,\mu}$ and $B_{\mu,\mu}$ are called the *reflection coefficients* and 0 is the zero matrix; all are of size $q \times q$. To solve for $\mathbf{F}_\mu$ in (9), we solve for the reflection coefficients in a

least squares sense, minimizing the squared residual error averaged over the sequence. The expected value of the reflection coefficients given the forward residual error is the same as the solution given the backward residual error [3]. However, averaging the two solutions is a more robust approach since we are dealing with a limited sample of the true sequence. We solve for reflection coefficients which minimize the weighted sum of the squared forward and backward residual errors averaged over the sequence, i.e.,

$$E_\mu \quad = \quad \sum_{t=\mu+1}^{\tau} \left[(e_{\mu,t})^T Q^f e_{\mu,t} + (b_{\mu,t})^T Q^b b_{\mu,t} \right], \tag{11}$$

where matrices Q^f and Q^b weight the impact of the forward and backward components. The relative accuracy of the lower-order forward and backward models provides confidence measures for current iteration. The higher the covariance of the driving distribution, the more uncertainty in the model and therefore the less confidence we have in the resulting estimates for the reflection coefficients. We set the weights to the inverse of the covariance of the driving distribution for the forward and backward models of order $M-1$, called the *power matrices*[1], i.e.,

$$Q^f = (P^f_{\mu-1})^{-1}, \qquad Q^b = (P^b_{\mu-1})^{-1}, \tag{12}$$

where,

$$P^f_{\mu-1} \quad = \quad \left[R_0 \;\; R_1 \;\; \ldots \;\; R_{\mu-1} \right] \mathbf{F}_{\mu-1}, \tag{13}$$

$$P^b_{\mu-1} \quad = \quad \left[R_{\mu-1} \;\; R_{M-2} \;\; \ldots \;\; R_0 \right] \mathbf{B}_{\mu-1}. \tag{14}$$

and R_i is the sample autocorrelation of the observed sequence under the assumption of stationarity,

$$R_i \quad = \quad \frac{1}{\tau-\mu} \sum_{t=\mu+1}^{\tau} x_t (x_{t-i})^T. \tag{15}$$

The power matrices are positive definite, and therefore invertible, in any physically realizable linear dynamic system [23]. Using nonsingular weight matrices provides a unique solution to the minimization of (11) [24]. Moreover, choosing such weights simplifies the equation significantly. By taking the derivative of E_μ with respect to the reflection coefficients $F_{\mu\mu}$ and using weights $P^f_{\mu-1}$ and $P^b_{\mu-1}$, the following is derived in [24],

$$HF_{\mu,\mu} + P^b_{\mu-1} F_{\mu,\mu} (P^f_{\mu-1})^{-1} D \quad = \quad -2G, \tag{16}$$

which one can use to solve for $F_{\mu,\mu}$. D and H are the covariance of the offset forward and backward residuals respectively, and G is the correlation between the offset residuals:

$$D = \sum_{t=1}^{\tau-\mu} \varepsilon_{\mu,t} (\varepsilon_{\mu,t})^T, \qquad H = \sum_{t=1}^{\tau-\mu} \beta_{\mu,t} (\beta_{\mu,t})^T, \qquad G = \sum_{t=1}^{\tau-\mu} \beta_{\mu,t} (\varepsilon_{\mu,t})^T. \tag{17}$$

[1] In the forward model shown in equation (4), WW^T is the power matrix.

The forward and backward offset residuals are defined as follows[2],

$$\varepsilon_{\mu,t} = x_{t+\mu} + \sum_{i=1}^{\mu-1} F_{\mu-1,i} x_{t+\mu-i}, \quad (18)$$

$$\beta_{\mu,t} = x_t + \sum_{i=1}^{\mu-1} B_{\mu-1,i} x_{t+i}. \quad (19)$$

We solve for $B_{\mu,\mu}$ using the generalized conjugate relationship [3],

$$B_{\mu,\mu} = (P^f_{\mu-1})^{-1} (F_{\mu,\mu})^T P^b_{\mu-1}. \quad (20)$$

From (9), (10), (13) , (14) , and (20), the following recursive updates can be derived for the power matrices [3],

$$P^f_M = P^f_{\mu-1} - (F_{\mu,\mu})^T P^b_{\mu-1} F_{\mu,\mu}, \quad (21)$$

$$P^b_M = P^b_{\mu-1} - (B_{\mu,\mu})^T P^f_{\mu-1} B_{\mu,\mu}. \quad (22)$$

Using this recursive definition, rather than (13) and (14), the higher-order autocorrelation estimates are not calculated from the sample sequence.

To initialize the algorithm, in the zero-order model we assume the sequence is the output of the stochastic component of the model. Therefore $P^f_0 = P^b_0 = R_0$, $\varepsilon_{0,t} = x_{t+1}$ and $\beta_{0,t} = x_t$.

To summarize MESA: Given the coefficients for a model of order $\mu - 1$, $\mathbf{F}_{\mu-1}$, and the state-space projection, X_1^τ, of the observed image sequence, (13) and (14) are used to determine the power matrices, $P^f_{\mu-1}$ and $P^b_{\mu-1}$, and (18) and (19) solve for the offset residuals, $\varepsilon_{\mu,t}$ and $\beta_{\mu,t}$. The forward reflection coefficients $F_{\mu,\mu}$, which minimize the squared sum of weighted residual errors, are determined by (16) and the backward reflection coefficients $B_{\mu,\mu}$ are calculated using the generalized conjugate relationship (20). Using the reflection coefficients $F_{\mu,\mu}$ and $B_{\mu,\mu}$, and the lower-order model parameters $\mathbf{F}_{\mu-1}$, (9) and (10) provide the coefficients $\mathbf{F}_\mu$ for a model of order μ.

4 Results

There are several ways one can evaluate and compare synthesized image sequences [1]. Here we use the one-step prediction error to quantify the quality of our results, as in [6],

$$err_\mu(i) = || y_i + C(\sum_{j=1}^{\mu} F_{\mu,j} (C^\diamond y_{i-j})) ||_2, \quad (23)$$

where $C^\diamond \equiv C^T (CC^T)^{-1}$ is the pseudo-inverse of C.

Higher-order dynamic models are shown to improve the average one-step prediction error for the test sequences in Fig. 1. As more temporal information is used to generate subsequent image frames, the prediction error of the synthesized images decreases.

[2]The notation for the indices of the offset residuals is somewhat counter-intuitive. Nonetheless, it is used throughout time-series literature and, for consistency, it will be used here as well. Residuals $\varepsilon_{\mu,t}$ and $\beta_{\mu,t}$ use the estimation from models of order $\mu - 1$, however a different interval of states is used within the calculation.

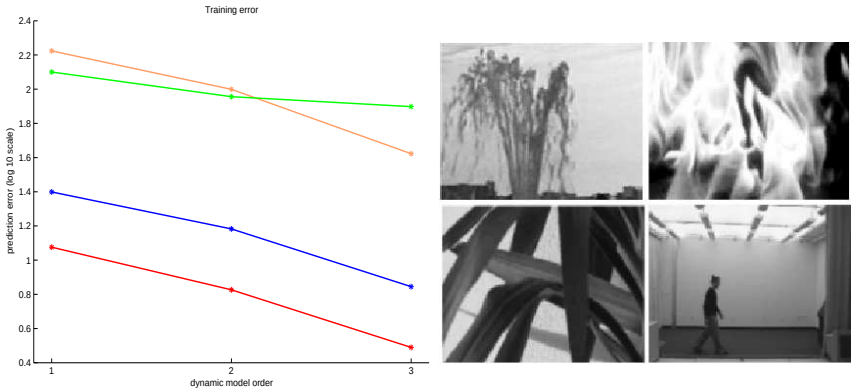


Figure 1: The effect of changing the order μ of the dynamic model is shown for four sequences: the fountain sequence [25] (blue), the fire sequence [25] (yellow), the house plant sequence [11] (green), and the walking sequence [19] (red). A frame of each sequence is shown on the right. The house plant sequence was trained with $\tau = 200$ frames and the others with $\tau = 80$. Appearance model consisted of 25-dimensions. The one-step prediction error was average over all $\tau - \mu$ sets of initialization frames.

Depending on the type of motion in the scene, the advantage of second and third-order dynamic models varies. In the house plant sequence the oscillatory swaying motion of the leaves is not captured by first-order dynamics but can be modeled using second-order dynamics. Third-order dynamics, however, do not provide much further improvement. This improvement is illustrated on the left in Fig. 2. The effect of changing the length of the sequence used for training the dynamical model, is also shown in Fig. 2. For each length the mean error was calculated from 20 models trained on different intervals of the original sequence. For each model, the median[3] of the one-step prediction error is calculated over 40 initialization intervals sampled from the original sequence at regular intervals.

Although convenient for optimization, the one-step prediction error alone is not sufficient for evaluating of the overall quality of a synthesized sequence. Without the ability to consider extended intervals of time, the stability of the system is not captured. Moreover, the mean-squared error does not measure perceptual quality. For example, increasing the dimension of the appearance model decreases the prediction error, but beyond some small dimension there was no difference in perceptual quality for most textured sequences.

We found a higher-order dynamical model to be necessary to capture pendulum-like movement, such as the swaying of the leaves in the house plant sequence. In the synthetic sequences generated by a first-order model the leaves flicker, whereas, the sequences generated by a second-order model capture the swaying motion. In order to explore this, one can analyze the temporal frequency of the image intensities; we expect the jittery motion to exhibit more power at high-frequencies than the swaying motion. A set of image positions were randomly sampled according to a uniform distribution. The image sequence was spatially blurred by a Gaussian and then measured at the sampled locations.

[3]The median was used to accomodate short intervals of frames and increase robustness to the few instances when the power matrix was ill-conditioned causing the error to explode.

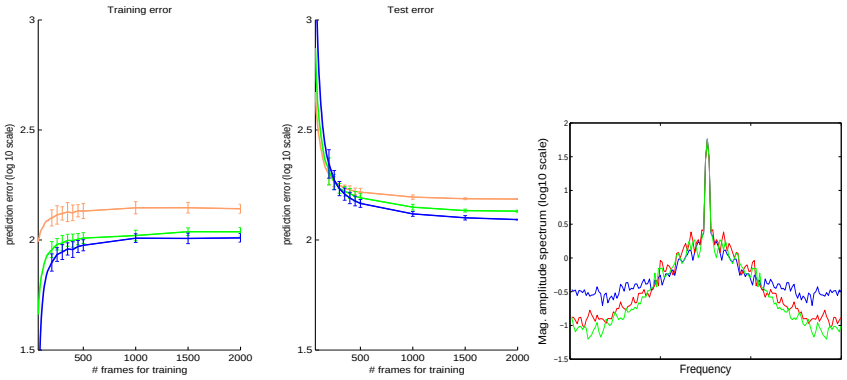


Figure 2: Results for the synthesized house plant sequence. LEFT: The effect of increasing the order of the dynamic model on house plant sequence syntheses. The one-step prediction error results reflect the visually observed results: increasing from a 1^{st}-order (yellow) to a 2^{nd}-order (green) dynamic model improves accuracy of the synthesized sequence more than increasing from a 2^{nd}-order to a 3^{rd}-order (blue) model. Models used appearance models of 25-dimensions and training lengths from 70-2000 frames. RIGHT: Average magnitude of the amplitude spectrum. The larger amount of high frequency information in the 1^{st}-order model (blue) is in loose agreement with the perception of the jittery motion in the video. The results from the 2^{nd}-order model (red) more closely resemble the training data (green).

After taking the Fourier transform of the resulting temporal signal, the magnitude of the frequency was averaged over all sampled positions to obtain one generalized signal for each synthesized sequence. A cosine temporal window was used before taking the Fourier transform to reduce windowing effects. The average amplitude spectrum for the first and second-order synthesized sequences of the house plant video are shown on the right in Fig. 2. The larger amount of high frequency information in the first-order model is in loose agreement with the perception of the jittery motion in the video.

When the autoregresssive model is provided with a sufficient number of frames for training, relative to MESA, the Yule-Walker method finds parameters which generate images with a smaller one-step prediction error in the first few frames. A full sequence cannot be generated using these parameters, however, because the predictions become inaccurate over time due to model instability. The stability of models learned with MESA is guaranteed, but the results of the model are not necessarily perceptually accurate. In particular, without a sufficient amount of training data, the power matrices are ill-conditioned and the error is significant. It is important to note, however, that neither the Yule-Walker method nor MESA will provide a useable model under such conditions.

4.1 Linear Model Limitations

Our results demonstrate that a significant amount of the movement in the scene can be captured with a linear autoregressive model, especially with higher-order dynamics. However, real-world visual scenes exhibit complex dynamics. As expected, there are nonlinear components within most observed motions which are not well described by our

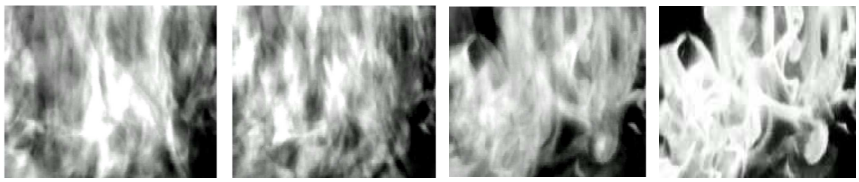


Figure 3: The higher-order dynamic models produce syntheses which resemble the original data over a longer interval. From left to right, frame 52 of the flame sequence syntehsized from $1^{st}, 2^{nd}, 3^{rd}$-order dynamic models, and the corresponding frame from the original sequence.

model.

The deterministic component of the dynamic model provides a linear prediction of the subsequent state and the final estimation lies within a multidimensional Gaussian distribution centered at this prediction. Similar images occupy a more complex manifold in the subspace and learning the manifold may require a lot of data to ensure a dense sample of the image space [17]. Using linear dynamics with a Gaussian driving distribution will not guarantee that predicted states remain on this manifold. Moreover, because the image dataset is not convex slight inaccuracies in the prediction cause dispersion artifacts in the synthesis. For example, in the fire sequence synthesis the flame filaments are distinct and compact initially, like the original sequence. As the length of the synthesis increases, the state predictions decrease in accuracy, drift further from the manifold and the flames are dispersed across the image plane. As the order of the model increases, however, the syntheses resemble the original data over a longer interval, as shown in Fig. 3.

5 Conclusion

The results of this work illustrate how higher-order dynamics contribute to the perceptual accuracy of the novel synthesized sequences generated by autoregressive models. The complicated motion patterns which extend over multiple frames of dynamic textures are more adequately represented when additional temporal information is provided during the learning process and when generating the motion in the scene.

Without sufficient training data, previously used techniques for learning autoregressive model parameters produced unstable and inaccurate results, in particular when using higher-order dynamic models. To overcome this limitation we applied MESA, a linear prediction technique common in control theory literature which generates a reliably stable autoregressive model.

Dynamic textured sequences are complicated scenes with complex motion patterns. We have found that a significant amount of the perceptually relevant information in the scene is captured by higher-order linear autoregressive models. The models explored in this work could be used either for an accurate prediction of a few frames ahead in the sequence or to capture a general description of the motion upon which more detail could potentially be incorporated. The latter opens an interesting direction for future research.

References

[1] See supplementary material; URL: http://www.cs.toronto.edu/~mhyndman

[2] Z. Bar-Joseph, R. El-Yaniv, D. Lischinski, and M. Werman. Texture mixing and texture movie synthesis using statistical learning. *Vis. Comp. Graph.*, 7(2):120–135, 2001.

[3] J. P. Burg. *Maximum Entropy Spectral Analysis*. PhD thesis, Stanford University, 1975.

[4] Antoni B. Chan and Nuno Vasconcelos. Probabilistic kernels for the classification of auto-regressive visual processes. In *IEEE CVPR*, 1:846–851, 2005.

[5] J. S. De Bonet. Multiresolution sampling procedure for analysis and synthesis of texture images. *Comp. Graphics*, 31:361–368, 1997.

[6] G. Doretto, A. Chiuso, Y. Wu, and S. Soatto. Dynamic textures. *IJCV*, 51(2):91–109, 2003.

[7] G. Doretto and S. Soatto. Editable dynamic textures. In *ACM SIGGRAPH Sketches*, 2002.

[8] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In *SIGGRAPH*, pp. 341–346, 2001.

[9] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. *IEEE ICCV*, 2:1033–1038, 1999.

[10] D. J. Heeger and J. R. Bergen. Pyramid-based texture analysis/synthesis. *SIGGRAPH*, pp. 229–238, 1995.

[11] Midori Hyndman. MSc Thesis, University of Toronto, 2006.

[12] B. Julesz. Visual pattern discrimination. *IRE Trans. Inform. Theory*, 8(2):84–92, 1962.

[13] M. Kaveh and G. Lippert. An optimum tapered burg algorithm for linear prediction and spectral analysis. *IEEE Trans. ASSP*, 31(2):438–444, 1983.

[14] V. Kwatra, A. Schodl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: image and video synthesis using graph cuts. *ACM Trans. Graph.*, 22(3):277–286, July 2003.

[15] L. Liang, C. Liu, Y. Xu, B. Guo, and H. Shum. Real-time texture synthesis by patch-based sampling. *ACM Trans. Graph.*, 20(3):127–150, July 2001.

[16] M. Morf, B. Dickinson, T. Kailath, and A. Vieira. Efficient solution of covariance equations for linear prediction. *IEEE Trans. ASSP*, 25(5):429–433, 1977.

[17] S. Nayar, H. Murase, and S. Nene. Parametric appearance representation. In *Early Visual Learning*, pp. 131–160. Oxford University Press, 1996.

[18] R. C. Nelson and R. Polana. Qualitative recognition of motion using temporal texture. *CVGIP: Image Underst.*, 56(1):78–89, July 1992.

[19] Georgia Tech. GVU Centre/College of Comp. *Human Identification at a Distance* website.

[20] J. Portilla and E. P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *IJCV*, 40(1):49–70, 2000.

[21] P. Saisan, G. Doretto, Y. Wu, and S. Soatto. Dynamic texture recognition. *IEEE CVPR*, 2:58–63, 2001.

[22] A. Schodl, R. Szeliski, D. Salesin, and I. Essa. Video textures. *SIGGRAPH*, pp. 489–498, 2000.

[23] G.A. Smith and A.J. Robinson. A comparison between the EM and subspace identification algorithms for time-invariant linear dynamical systems. Tech. Report, Cambridge Univ., 2000.

[24] O. Strand. Multichannel complex maximum entropy (autoregressive) spectral analysis. *IEEE Trans. on Automatic Control*, 22(4):634–640, 1977.

[25] M. Szummer and R. W. Picard. Temporal texture modeling. *ICIP*, 3:823–826, 1996.

[26] P. Van Overschee and B. De Moor. N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1):75–93, 1994.

[27] L. Wei and M. Levoy. Fast texture synthesis using tree-structured vector quantization. *SIGGRAPH*, pp. 479–488, 2000.

[28] Y. Xu, B. Guo, and H. Shum. Chaos mosaic: Fast and memory efficient texture synthesis. Techn. Report, Microsoft Research, April 2002.

Refining implicit function representations of 3-D scenes

Matthew Grum and Adrian G. Bors

Dept. of Computer Science, University of York, York YO10 5DD, UK
{grum,adrian.bors}@cs.york.ac.uk

Abstract

This paper considers the problem of modelling a 3-D scene from calibrated images taken from multiple viewpoints. The initial 3-D information is acquired using probabilistic space carving which provides a voxel representation consistent with the given set of images. The scene is afterwards modelled as an implicit surface using radial basis functions (RBF). The mixture of multiorder basis functions models a smoothed 3-D scene representation while providing compactness. We use correspondences between pairs of image patches in order to update the RBF centres for improving the 3-D scene representation. The RBF centre updating leads to improving the consistency between the 3-D model and the given set of images. The proposed method is applied on a complex 3-D scene displaying various objects.

1 Introduction

Three dimensional object reconstruction from several images has lately attracted considerable research interest [1, 8, 9, 12]. Nevertheless, real scenes are very complex and involve several objects, usually occluding each other, while the effects of illumination and material reflectivity cannot be ignored. The aim of this study is to reconstruct the entire 3-D scene from a sparse set of images by estimating both shape and texture.

Space carving is a method which assigns voxels to a 3-D object or to its background using the photoconsistency of a specific point with all its corresponding pixels from the given set of images [9, 10, 12]. There is a lot of uncertainty in the evaluation of the probabilities required for space carving, caused by the presence of occlusions, surface discontinuities, variation in the illumination conditions, camera calibration errors, etc. The resulting voxel model from space carving is invariably noisy and often contains disconnected components. Holes and excessively enlarged 3-D features emerge in the resulting voxel model [9]. Surface refinement for mesh models initialised from volumetric reconstruction has been performed in [5, 6]. In this paper we propose to employ a radial basis function (RBF) in order to model the surface of the space carved data. RBF methods are known for their data fitting, interpolation and generalisation properties and have been widely used in pattern recognition. In our case we want to represent a smooth surface which interpolates the voxels from the surface as accurately as possible to the real scene. Moreover, the RBF model would require only few parameters when compared to the voxel model in order to represent the scene. Implicit RBFs have been shown to represent well surfaces in [4, 11].

In this paper we use the multiorder basis function, proposed by Chen and Suter, which fulfils a smoothness constraint in the first, second and third order Laplacian [2]. The surface of the objects from the scene is calculated as the zero level set of a weighted mixture of basis functions. The basis functions centres are randomly initialised by using a Poisson sphere random sampling scheme [3, 4].

Certain errors are propagated from the voxel model to the implicit surface resulting in surface variations that do not correspond to the actual scene. In this paper we propose to correct such errors by improving the consistency of the 3-D model with the given set of images. In order to achieve good reconstruction accuracy we need to select a wide baseline pair of images with good texture. The pair of images contain the projection of the same part of the 3-D scene, defined around radial basis function centres. An updating formula is derived such that the centre of a certain RBF unit is modified in order to fulfil the consistency between the two projections of the 3-D scene. The proposed methodology is applied on a complex scene representing several objects. The modelling of a 3-D scene using space carving and the modelling using RBF is described in Section 2. The initialisation of the RBF parameters as well as their subsequent updating is described in Section 3. Experimental results are provided in Section 4, while the conclusions of this study are drawn in Section 5.

2 Model initialisation

2.1 Space carving

Let us assume that we have N images of a scene $\{\mathbf{I}_j | j = 1, \ldots, N\}$, acquired from various viewpoints by calibrated cameras whose projective matrices $\mathbf{P}_j$ with respect to the scene have been properly calculated. We would like to reconstruct the 3-D scene represented by geometry as well as colour (texture) information. One of the most popular approaches for representing 3-D scenes from multiple images is the space carving algorithm [1, 10, 12]. Probabilistic space carving starts with a parallelepiped formed from voxels. At each iteration, a set of voxels is selected and their consistency with the given set of images is verified. Two assumptions are tested: if a voxel is part of the scene $\mathbf{x} \in \mathcal{V}$, and if it is not, $\mathbf{x} \notin \mathcal{V}$, where $\mathbf{x}$ represents a voxel and $\mathcal{V}$ is the volumetric scene to be estimated. The evaluation of the probability in each image $\mathbf{I}_j$ takes into account its corresponding projection matrix $\mathbf{P}_j$ and checks the photoconsistency of a voxel with its corresponding pixels. Usually, uncertainty arises in the evaluation of the probabilities associating voxels with corresponding pixels from images. Consequently, the resulting volumetric model is invariably noisy. Esteban and Schmitt [5] proposed to use the visual hull in order to initialise a surface mesh which can be deformed under the influence of photoconsistency constraints. In the following we propose to use implicit function modelling estimated from the 3-D voxel data provided by the space carving algorithm.

2.2 Implicit surfaces using radial basis functions

Radial basis functions (RBF) are known for their data fitting, interpolation and generalisation properties [4, 11]. In our approach we use the voxel representation provided by the space carving algorithm by properly interpolating the voxels and smoothing the surfaces in the scene. Moreover, an RBF model would require fewer parameters in order to

represent the scene. The surface of the 3-D scene is modelled as a zero level set of a function, $f(\mathbf{z}) \geq 0$. In our approach, $f(\mathbf{z})$ is an RBF mixture consisting of M basis functions calculated at location $\mathbf{z}$ as :

$$f(\mathbf{z}) = \sum_{i=1}^{M} w_i \phi(\|\mathbf{z} - \mu_i\|) + u(\mathbf{z}) \tag{1}$$

where $\phi(\cdot)$ is the basis function, considered radially symmetric, $\|\cdot\|$ is the Euclidean distance, μ_i is the basis function centre, and $u(\mathbf{z})$ is a polynomial component. The function $f(\mathbf{z})$ is defined as positive inside the 3-D volume and negative outside. For $f(\mathbf{z}) = 0$ we obtain the surface enveloping the 3-D voxel model.

Gaussian RBF functions which are widely used in pattern recognition have been found to oversmooth [4]. Chen and Suter derived a basis function which fulfils a constraint in the first, second and third order Laplacian, [2] :

$$-\delta\Delta f(\mathbf{z}) + \Delta_2 f(\mathbf{z}) - \tau\Delta^3 f(\mathbf{z}) = 0 \tag{2}$$

where Δ is the Laplacian operator in 3-D, δ is a parameter controlling the first order smoothness and τ controls the third order smoothness. The function that minimises the energy function from (2) is called multiorder basis function [2, 4] :

$$\phi(\|\mathbf{z} - \mu_i\|) = \frac{1}{4\pi\delta^2\|\mathbf{z} - \mu_i\|}\left(1 + \frac{\beta e^{-\|\mathbf{z}-\mu_i\|\sqrt{\alpha}}}{\alpha - \beta} - \frac{\alpha e^{-\|\mathbf{z}-\mu_i\|\sqrt{\beta}}}{\alpha - \beta}\right) \tag{3}$$

where

$$\alpha = \frac{1 + \sqrt{1 - 4\tau^2\delta^2}}{2\tau^2} \;;\; \beta = \frac{1 - \sqrt{1 - 4\tau^2\delta^2}}{2\tau^2} \tag{4}$$

are parameters which describe the shape of the basis function.

3 RBF parameter calculation

3.1 Initialising the RBF parameters

The RBF function has the property to approximate well the data in a specific neighbourhood as shown by the expression (3). The RBF function from (3) has the maximum in the centre μ_i and quickly falls toward zero when the distance from its centre location increases. In this study we use the Poisson sphere random sampling scheme for initialising the RBF centres [4]. This algorithm has been proposed in [3] by Cook for solving the aliasing problem in computer graphics. A Poisson sphere distribution is a 3-D random point distribution in which all sphere centres are approximately equally distributed in space. Let us consider a set of spheres as $\mathbf{S}(\mu_k, \rho)$, $k = 1, \ldots, M$, each centred at μ_i and with identical radius, ρ. The sphere radius ρ depends on the size of the voxel model, $|\mathscr{V}|$. The number of basis functions M and consequently that of spheres depends on the desired level of surface approximation and smoothness.

Centres of spheres are randomly generated within the given voxel space such that they fulfil the following conditions :

$$\|\mu_i - \mu_j\| \geq 2\rho \tag{5}$$

where 2ρ is the minimum distance between two sphere centres i and j. Each sphere determines a partition in the voxel model depending on the local compactness. Let us

consider a set of at least T connected voxels which are located within a radius of ρ from the centre of the sphere :

$$\{\mathbf{x}_c \in \mathbf{S}(\mu_i,\rho) | \mathbf{x}_c \in \mathscr{V}, \|\mathbf{x}-\mu_i\| < \rho, |\mathbf{x}_c| \geq T\} \tag{6}$$

where $|\cdot|$ denotes set cardinality. The spheres which contain very few voxels as well as unconnected voxels are discarded. The sphere generating algorithm terminates when the voxel model is completely covered with spheres. Let us assume that a total of M valid spheres $\mathbf{S}(\mu_i,\rho)$ are generated, each associated with an RBF centre, μ_i. The parameters τ and δ determine the smoothing of the resulting implicit surface. These parameters are chosen depending on the chosen resolution, the size of the voxel model $|\mathscr{V}|$, and on the desired level of smoothing [4].

We form the following system of equations :

$$\begin{bmatrix} \phi(r_{11})+\lambda_1 & \dots & \phi(r_{1M}) & 1 \\ \vdots & \ddots & \vdots & \mathbf{1} \\ \phi(r_{M1}) & \dots & \phi(r_{MM})+\lambda_M & 1 \\ 1 & \mathbf{1} & 1 & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_M \\ u_0 \end{bmatrix} = \begin{bmatrix} f(\mu_1) \\ \vdots \\ f(\mu_M) \\ 0 \end{bmatrix} \tag{7}$$

where $r_{ij} = \|\mu_i - \mu_j\|$ is the Euclidean distance between two centres, λ_i for $i = 1,\dots,M$ are added to the diagonal elements in order to condition better the matrix as in [4] and $u(\mathbf{z}) = u_0$. For calculating the weighting factors w_i, $i = 1,\dots,M$ we evaluate the basis functions $\phi(\cdot)$ for the distances between pairs of centres r_{ij}. We consider $f(\mu_i) = 0$ for imposing the condition that most basis functions are located on the separation surface. Certain centres correspond to the control basis functions, *i.e.* which have their centres either inside the model or outside it. The weights w_i, $i = 1,\dots,M$ are calculated by inverting the matrix associated with basis function centres. Given the proposed RBF centre initialisation described in the previous Section, the matrix from (7) is non-singular and consequently invertible.

3.2 Updating the RBF centres using image disparity

The previous approaches adopted in space carving have been restricted to considering per voxel consistency. In this Section we describe how the accuracy of the surface can be improved by considering the image consistency across larger areas of the surface. Invariably, given various sources of errors, the surface described by the implicit function $f(\mathbf{z})$ may not fit with its corresponding areas of the images. In this Section we describe how to find an updating transformation applied on the basis function parameters in order to improve the consistency between the 3-D model given by $f(\mathbf{z})$ and the image set $\{\mathbf{I}_j | j = 1,\dots,N\}$. The surface, as defined by (7), passes through the radial basis function centres. We can control the surface by changing the locations of the RBF centres. The first order approximation of an RBF consists of the plane tangent to its surface in the neighbourhood of its centre. For a small area well defined around the RBF centre we assume that the surface function $f(\mathbf{z})$ can be locally approximated by a planar patch. Let us assume that there are two images which contain projections of the 3-D patch. A plane in 3-D, such as the one which approximates locally the surface around the basis function centre, induces a homography $\mathbf{H}$ between pairs of images [8]. By calculating $\mathbf{H}$ from pairs of images it is possible to recover the parameters of this plane and correct the basis function centre and thus that part of the surface, by constraining it to lie on that plane.

A surface patch, corresponding to an RBF centre, is selected if it displays a sufficient amount of detail which can be used for finding matches between pairs of images. For each chosen patch we select a pair of images such that they provide the smallest angle between their positions and the surface patch normal. The angle between the camera locations and the surface patch should be as large as possible in order to provide an appropriate baseline to recover positions. Let $\mathbf{P}$ and $\mathbf{P}'$ be two 3×4 matrices which describe the camera projection from 3-D coordinates to homogenous image coordinates for the selected pairs of images representing the patch. Let $\mathbf{y} = [u, v, 1]^T$ be the projection of a point in the patch from the first camera and $\mathbf{y}' = [u', v', 1]^T$ be the corresponding point from the second camera. These points are related by $\mathbf{y}' = \mathbf{H}\mathbf{y}$. Let us assume that the selected patch belongs to a plane ψ, where $\mathbf{z}^T\psi = 0$ for all the points $\mathbf{z}$ which lie on ψ. The homography $\mathbf{H}$ between the pair of images is given as, [8] :

$$\mathbf{H} = \mathbf{A} - \mathbf{a}\mathbf{v}^T \tag{8}$$

where $\mathbf{A}$ and $\mathbf{a}$ are a 3×3 matrix and a 3×1 vector, respectively, given by :

$$[\mathbf{A} \mid \mathbf{a}] = \mathbf{P}' \begin{bmatrix} \mathbf{P} \\ 0\,0\,0\,1 \end{bmatrix}^{-1} \tag{9}$$

and $\mathbf{v}$ is a 3×1 vector, representing the displacement between the two images, given by the following expression :

$$\begin{bmatrix} \mathbf{v} \\ 1 \end{bmatrix} = \left(\begin{bmatrix} \mathbf{P} \\ 0\,0\,0\,1 \end{bmatrix}^{-1} \right)^T \psi \tag{10}$$

Given a point in one image, the corresponding point in another image can be constrained to lie on a line known as the epipolar line [8]. Epipolar lines depend only on the imaging geometry and not on the shape of the scene, so it is possible to transform the images, using $\mathbf{v}$, in order to correspond to a pair of rotated 'virtual cameras', whose epipolar lines are all horizontal and co-linear. This process is known as rectification and is often performed as an initial step in stereo algorithms [7].

Let $\mathbf{R}$ and $\mathbf{R}'$ be the rectifying 3×3 matrix transformations. The rectified images of the patch are now related by considering a matrix $\mathbf{H}_R$:

$$\mathbf{R}'\mathbf{y}' = \mathbf{H}_R\mathbf{R}\mathbf{y} \tag{11}$$

The homography $\mathbf{H}$ can be calculated by taking into account the rectifying transformations :

$$\mathbf{H} = \mathbf{R}'^{-1}\mathbf{H}_R\mathbf{R} \tag{12}$$

After the rectification, the epipoles are horizontal, and $\mathbf{H}_R$ is guaranteed to map each v-coordinate to its corresponding value in each pair of images. Consequently, it can be expressed as :

$$\mathbf{H}_R = \begin{bmatrix} s & k & t \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{13}$$

where s, k and t, correspond to scaling, skew and translation, respectively (all in the u direction). To calculate these parameters, the images of the patch are divided into l rows

of pixels. When considering a single row of pixels, the skew and translation act together to produce a single horizontal offset, o, since the v coordinate of each pixel is the same.

The normalised cross-correlation is computed between each pair of rows at different scale and offset values. The values which result in the lowest score, corresponding to the best match, are recorded. As the scale should be the same for all rows, s is taken to be the median of the values found for each row. Any values significantly outside the median are deemed to be unreliable and are discarded. Using the offsets from all rows, the skew and translation parameters k and t can be calculated by solving a linear system:

$$\begin{bmatrix} k \\ t \end{bmatrix} = \begin{bmatrix} v_1 & 1 \\ \vdots & \vdots \\ v_l & 1 \end{bmatrix}^{-1} \begin{bmatrix} o_1 \\ \vdots \\ o_l \end{bmatrix} \tag{14}$$

where v_l is the v coordinate of row l.

With $\mathbf{H}$ at hand from (12), we calculate the displacement vector $\mathbf{v}$ between the pair of images corresponding to the given patch from (8). Consequently, the location of the plane ψ which should contain the basis function is calculated using (10). The location of the basis function centre is updated as :

$$\mu_i' = \mu_i + \mathbf{n}\frac{\mu_i^T \psi}{\mathbf{n}^T\mathbf{n}} \tag{15}$$

where $\mathbf{n}$ is the surface normal direction of the plane ψ, and the ith basis function centre μ_i is updated to μ_i', while being constrained to lie on the plane ψ.

The matching based on cross-correlation requires that the colour variance in the image patch is above a certain threshold in order to find the offsets uniquely. Basis functions corresponding to patches which do not fulfil this condition are not updated by this procedure. Additionally, false matches may be obtained due to the image noise or to patches which span the physical boundary of an object. A limit is placed on the maximum distance that a centre can move in order to prevent this from causing further errors in the surface. Some of the basis function centres will converge towards neighbouring locations on the 3-D surface causing singularity in the matrix from equation (7). If centres of multiple basis functions occur in the immediate proximity of each other after updating, only one will be preserved while the others will be removed.

4 Experimental results

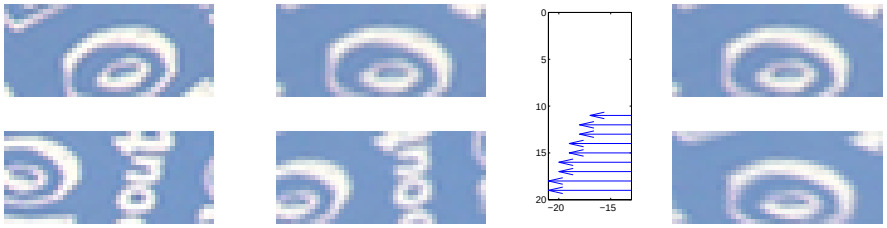
The method outlined in this paper was tested on a real scene comprised of multiple objects. For the experiments, 12 images of the scene were captured from various viewpoints. A selection of four images is shown in Fig. 1. As it can be observed from this Figure, the objects exhibit various shapes and surface properties and occlude each other in different views. Voxel carving assumes the camera positions (extrinsic calibration) to be known *a priori*. Targets printed on rectangular boards were placed around the outside of the scene in order to provide the necessary information for camera calibration.

The initial voxel model was provided by the probabilistic space carving algorithm [1]. This algorithm assumes the scene to be contained within a finite bounding volume. In this case, the background was manually segmented. The resulting model, which contains 773660 voxels, is shown in Fig. 4(a) and Fig. 4(b) for two different viewpoints. Considering the voxel representation, 4440 radial basis functions were sampled and used to

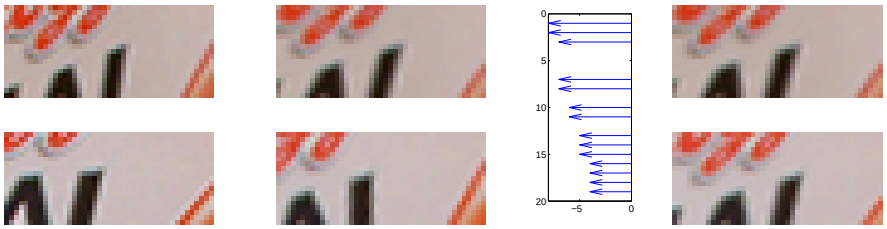
(a) Frame 1 (b) Frame 3

(c) Frame 6 (d) Frame 9

Figure 1: Four images of a complex scene taken from various viewpoints.

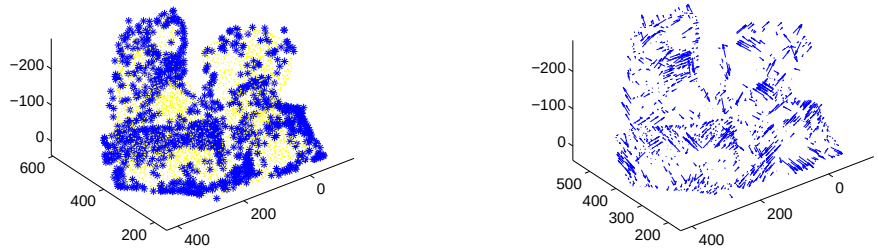


(a) Initial pair of projected book patches. (b) Epipolar rectification given by $\mathbf{H}_R$. (c) Offset vectors. (d) Aligned projected patches



(e) Initial pair of projected box patches. (f) Epipolar rectification given by $\mathbf{H}_R$. (g) Offset vectors. (h) Aligned projected patches

Figure 2: Two pairs of patches and their correction.



(a) RBF centres after updating. Updated centres are marked by "*".

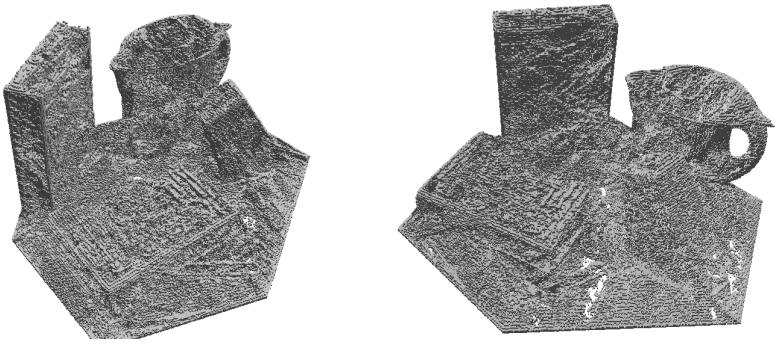
(b) RBF centres correction vectors.

Figure 3: Directions of centre updating and their corrected positions.

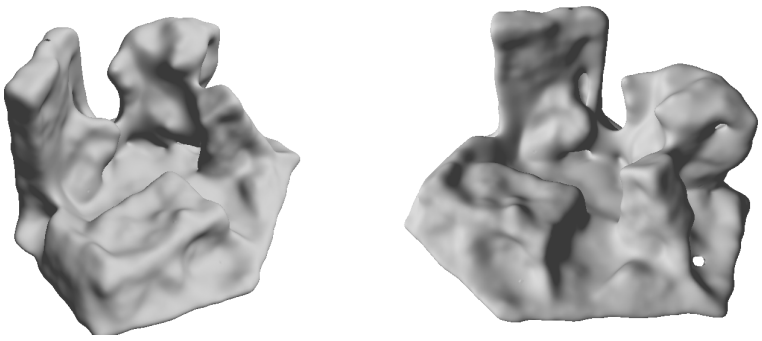
fit the implicit surface as shown in Figs. 4(c) and 4(d). A total of 1408 basis functions met the criteria for updating (the visibility and the presence of sufficient local variation as given by the colour variance). Of these, a suitable match was obtained in 1075 cases. The smoothing parameters, considered identical for all RBFs, are $\delta = 25$, $\tau = 0.01$, while the centres are scaled such that they fit in a cube of size $1 \times 1 \times 1$. The functions which were successfully updated are shown in Fig. 3(a), marked with stars, while all the other basis functions are marked with dots.

Two pairs of patches from the raw images, which are the projections of two different 3-D scene regions, one corresponding to the book and another to the box, are shown in Figs. 2(a) and 2(e), respectively. The images from each pair are related by means of $\mathbf{H}$, according to (8). The epipolar correction, as given by $\mathbf{H}_R$ from (13), is shown in Figs. 2(b) and 2(f), the offset vectors calculated from equation (14) are provided in Figs. 2(c) and 2(g), and the aligned patches after applying the transformation $\mathbf{H}$ to the second image of each pair is illustrated in Figs. 2(d) and 2(h). Vectors representing the movement of centres in 3-D to the corrected positions, according to (15), are shown in Fig. 3(b). The updated surface, after correcting the RBF centres, for the two viewpoints, is shown in Fig. 4(e) and Fig. 4(f), respectively. The surface of the horizontal book and vertical box is clearly improved. However, the shape of certain objects, the kettle in particular, is not well modelled due to their irregular shapes, lack of texture and surface specularity.

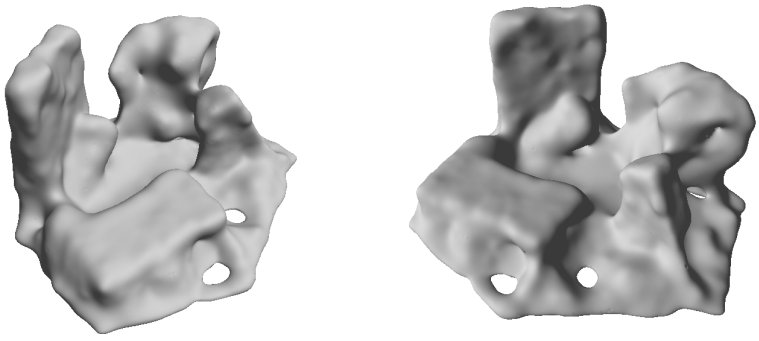
For numerical assessment we check the consistency between the surface of the book from the estimated 3-D model with that from the real scene. The surface of the book in the centre of the scene is planar and we measure the deviation from the planarity in the estimated 3-D model. This deviation, measured in millimetres, was estimated for the voxel model, the initial RBF surface, calculated according to the description from Section 3.1 as well as for the surface updated according to the algorithm provided in Section 3.2. The mean deviation was found to be 11.59 mm for the voxel model, 3.63 mm for the initial RBF estimation and 1.04 mm for the updated model. These numerical results together with the visual interpretation from Fig. 4 prove the capabilities of the proposed algorithm to improve the surface representation when considering the proposed RBF centre updating method based on image disparity estimation.



(a) (b)



(c) (d)



(e) (f)

Figure 4: Voxel representation and RBF surface modelling of the scene for two different view angles. (a), (b) voxel representation; (c), (d) initial RBF model; (e), (f) updated RBF model.

5 Conclusion

A complex 3-D scene surface modelling method using multiple images, taken from various viewpoints, is proposed in this paper. A voxel representation is estimated using the space carving algorithm. Implicit multiorder radial basis functions are employed in order to model the separation surface between the voxel model and the exterior. The RBF model produce a smoother 3-D scene than the voxel representation while requiring much less parameters. The 3-D representation is improved by using an RBF centre updating algorithm. The proposed algorithm estimates the disparity errors between pairs of images after recovering their perspective distortions. The resulting 3-D surface representation can be easily rendered and manipulated by geometrical transformations.

References

[1] A. Broadhurst, T. W. Drummond, and R. Cipolla. A probabilistic framework for space carving. In *Proc. ICCV, vol. 1*, pages 388–393, Vancouver, BC, Canada, 2001.

[2] F. Chen and D. Suter. Multiple order Laplacian spline - including slines with tension. Technical Report MECSE 1996-5, Monash University, Australia, July 1996.

[3] R. L. Cook. Stochastic sampling in computer graphics. *ACM Transactions on Graphics*, 5(1):51–72, 1986.

[4] H. Q. Dinh, G. Turk, and G. Slabaugh. Reconstructing surfaces by volumetric regularization using radial basis functions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(10):1358–1371, 2002.

[5] C. H. Esteban and F Schmitt. Silhouette and stereo fusion for 3D object modeling. *Computer Vision and Image Understanding*, 96(3):367–392, 2004.

[6] Y. Furukawa and J. Ponce. Carved visual hulls for image-based modeling. In *Proc. ECCV, part I, LNCS 3951*, pages 564–577, Graz, Austria, 2006.

[7] A. Fusiello, E. Trucco, and A. Verri. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12(1):16–22, 2000.

[8] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, 2004.

[9] H. Kim and I. S. Kweon. Appearance-cloning: photo-consistent scene recovery from multi-view images. *Int. Journal of Computer Vision*, 66(2):163–192, 2006.

[10] K. Kutulakos and S. M. Seitz. A theory of shape by space carving. *Int. Journal of Computer Vision*, 38(3):198–218, 2000.

[11] B. Scholkopf, J. Giesen, and S. Spalinger. Kernel methods for implicit surface modeling. In *Proc. of Neural Information Proc. Systems (NIPS)*, pages 1193–1200, 2004.

[12] G. G. Slabaugh, W. B. Culbertson, T. Malzbender, M. R. Stevens, and R. W. Schafer. Methods for volumetric reconstruction of visual scenes. *Int. Journal of Computer Vision*, 57(3):179–199, 2004.

Structure from Motion via a Two-Stage Pipeline of Extended Kalman Filters

Brian Clipp
bclipp@cs.unc.edu

Gregory Welch
welch@cs.unc.edu

Jan-Michael Frahm
jmf@cs.unc.edu

Marc Pollefeys
marc@cs.unc.edu

Department of Computer Science
University of North Carolina at Chapel Hill
Chapel Hill, NC USA

Abstract

We introduce a novel approach to on-line structure from motion, using a pipelined pair of extended Kalman filters to improve accuracy with a minimal increase in computational cost. The two filters, a *leading* and a *following* filter, run concurrently on the same measurements in a synchronized producer-consumer fashion, but offset from each other in time. The leading filter estimates structure and motion using all of the available measurements from an optical flow based 2D tracker, passing the best 3D feature estimates, covariances, and associated measurements to the following filter, which runs several steps behind. This pipelined arrangement introduces a degree of non-causal behavior, effectively giving the following filter the benefit of decisions and estimates made several steps ahead. This means that the following filter works with only the best features, and can begin full 3D estimation from the very start of the respective 2D tracks. We demonstrate a reduction of more than 50% in mean reprojection errors using this approach on real data.

1 Introduction

Structure from motion (SfM) is a well studied problem in computer vision. Most approaches begin with a set of salient 2D image features that are tracked from frame to frame using optical flow or wide baseline feature matching. Feature selection, determining which features to use in the structure from motion, is critical to the accuracy of results. Common approaches include RANSAC [8], robust regression [11] and filtering approaches which use a camera motion model to determine outliers in systems using Kalman or particle filter based 3D trackers [6, 7].

Our novel approach combines two extended Kalman filters that run concurrently on the same measurements in a synchronized producer-consumer fashion, but offset from each other in time. The leading filter generates initial estimates of sparse scene structure and the camera motion by identifying 2D tracks called *inliers* in the total set of 2D feature tracks. The subset of inliers determined by the leading filter provide better information about the camera pose. The leading filter passes their 3D estimates and covariances

(which have been improved by the influence of many measurements) to the following filter, which operates only on these good feature tracks with reliable initial 3D estimates.

In the experimental evaluation we demonstrate our pipelined approach on real data where it reduces the reprojection errors of the estimated 3D points in the following filter by more than 50%. This reduction in reprojection error reflects the fact that the pipelined two-filter approach only employs measurements that have been found to be consistent with the camera motion in the immediate future. While the improved estimates are delayed in time compared to the newest frame, which might be a concern for on-line applications, the approach allows the user to trade off this delay for improved performance.

In contrast to our approach which uses all temporal correspondence information over multiple frames (chains of matches), typical previous SfM approaches only employ correspondences from a single pair of frames. This is a result of the correlation of their computational cost with the probability of correct correspondences. As the probability of a chain of correspondences is significantly lower than for a single correspondence, previous approaches are often not efficient on chains of correspondences. (For a more complete overview of robust estimation in computer vision we suggest [10].) Our approach is efficient in that a naive approach to looking ahead w frames for inliers would run with $O(wh)$ complexity where h is the cost of one complete structure from motion estimation over all of the frames, while our two-stage filtering approach requires only $O(h)$ time.

In the next section we will discuss work related to the pipelined filter. Section 3 describes the pipelined filter architecture in detail and section 4 presents some experimental results that demonstrate the improvement in reprojection errors by our two-stage (leading-following) pipelined multi-filter approach.

2 Related Work

A key component of any structure from motion system is the estimation of the camera motion in 3D space from 2D feature tracks. Typically the obtained tracks contain a fair number of outliers. Hence the estimator has to simultaneously estimate the camera motion and to classify the tracks into outliers and inliers. Robust estimators are successfully applied to solve this problem in many computer vision applications. The most common technique to deal with outliers is the RANSAC algorithm [8, 20]. It solves the two problems of computing a relation that best fits the data and classifying the data as inliers (correct matches) and outliers. The classification is done by employing a cost function together with a threshold which depends on the expected measurement noise. The relation is then selected as the one with the highest number of inliers or the largest robust likelihood [8]. An inlier with respect to an error function has an error less than a threshold.

When the expected noise is not known beforehand it is difficult to determine the appropriate threshold. Then often robust regression methods are used to estimate the relation of the images and the classification of the data into inliers and outliers [11]. These methods achieve the greatest success when the data belong to a single signal corrupted with random outliers. Miller and Stuart [12] extended the MINPRAN robust regression method [18] to account for data that belong to multiple signals. The MINPRAN operator [18] tolerates a large number of outliers and identifies regions composed completely of outliers.

Tang et al. [19] proposed a tensor voting based approach that poses the problem of estimating the epipolar geometry (the focus of the paper which can be extended to many other

estimation problems) as one of finding the most salient hyperplane in a multi-dimensional space. Another popular technique is the Least Median of Squares (LMS) estimation [16]. LMS has been very successful when applied to a lone signal corrupted with outliers but fails completely if the outlier rate is higher than 50%. LMS searches a space of hypothesized fits using an objective function based on the median squared residual.

Another class of estimators adds a camera motion model to assist in detecting outliers. This measurement selection approach is based on a smooth motion model and consensus and is used with a Kalman filter in [1, 3] and a real-time particle filter in [7]. Davison presents a real time extended Kalman filter based visual simultaneous localization and mapping (SLAM) system in [6]. He uses a top down approach to measurement selection, searching for 2D features only in the region they are expected to be in the image based on estimation uncertainty, to minimize computational cost per frame.

Finally, we note that ideas for *fixed point* and *fixed lag* smoothing within a single Kalman filter were introduced by Rauch et al. [13, 14, 15]. The basic idea is to recursively estimate the state at some past time, either at some particular point in time, or following the current time with a fixed delay, using all of the available measurements. By separating the estimation into two pipelined filters we are able to prevent outliers from negatively affecting the second (following) filter, while simultaneously providing the following filter with non-causal initial estimates of the 3D points and covariances. In effect, we obtain some of the benefits of fixed-lag smoothing, but using only the inliers.

3 Two-Stage Measurement Selection and Estimation

The two-stage measurement selection/initialization and final estimation 3D pose filter is composed of two individual extended Kalman filters (EKF). We refer to them as the *leading* and *following* filters. The two filters run concurrently on the same measurements (images) in a synchronized producer-consumer fashion, but offset from each other in time. They are identical except in the way that they initialize 3D feature estimates, where the leading filter initializes points by triangulation and the following filter receives its initial feature position and covariance estimates from the leading filter.

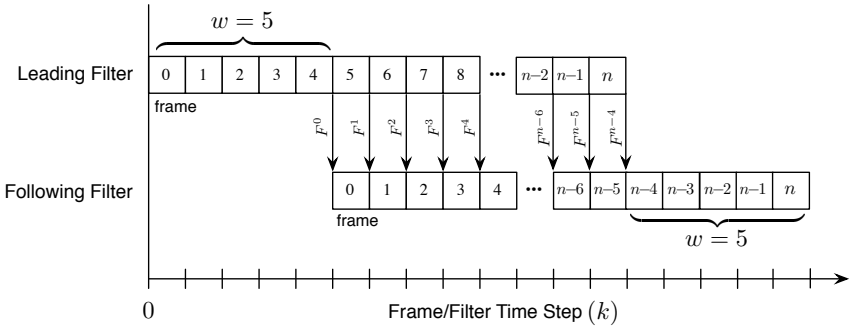


Figure 1: Timing of pipelined leading and following filters for time offset $w = 5$.

Figure 1 shows an example of pipelined leading and following filters for time offset $w = 5$. Once the initial w frames have been processed (the pipeline primed) then at each

filter time step k the leading filter passes its latest feature set F^{k-w} for frame $k-w$ to the following filter. Omitting the w for clarity, the feature set F^k for frame k is defined as

$$F^k = \{(x_1^k, \hat{X}_1^k, \Sigma_1^k), \ldots, (x_n^k, \hat{X}_n^k, \Sigma_n^k)\} \tag{1}$$

where x_i^k is the actual measurement (2D projection) of a feature, $\hat{X}_i^k$ is the *estimated* 3D position that feature at time k, Σ_i^k is the corresponding 3D covariance of the estimate, and n is the total number of features. The leading filter spends w time steps attempting to estimate 3D feature locations for frame $k-w$, selecting only the best ones to pass on to the following filter. In the remainder of this section we will first describe a single filter and then describe how the two filters are combined to form the estimation system.

3.1 Individual Filter

While we use an extended Kalman filter for this work, we believe the pipelined approach could be employed with any on-line 3D filters or other estimators. Our filters fuse measurements from a 2D KLT tracker [9, 17], which is an optical flow based 2D tracker that measures the motion of salient features from one frame to the next in a video sequence. The filter's process model uses a smooth motion model for the change in camera position and orientation. It uses a first order Taylor series approximation to relate the state at time k to time $k+1$. This model assumes that the velocity is constant. The estimated 3D features must be static with respect to the world frame to be included in the filter state, and so they are modeled as having zero velocity.

The filter state S^k at time k is made up of the camera's position C^k, orientation θ^k, velocity $\dot{C}^k$, orientation rate $\dot{\theta}^k$ (rotational velocity) and estimates of the 3D position of each of the n features being tracked $X_1^k ... X_n^k$. The filter state is shown in Equation (2),

$$S^k = \begin{bmatrix} C^k & \dot{C}^k & \theta^k & \dot{\theta}^k & X_1^k & \ldots & X_n^k \end{bmatrix}^T \tag{2}$$

where again, n is the number of tracked features. The filter's predicted measurement equation is simply the projection of each estimated 3D feature i into the camera at time k given calibration K. In Equation (3) R is the rotation matrix composed from the Euler angle representation of the camera orientation Θ^k.

$$\hat{x}_i^k = K \begin{bmatrix} R^{T^k} & -R^{T^k} C^k \end{bmatrix} X_i^k \tag{3}$$

Note that we use the "hat" in $\hat{x}_i^k$ to indicate it is an *estimate* of the measurement x_i^k.

To predict the actual measurement the projected 3D point must be homogenized, which requires dividing by the third homogeneous coordinate. This makes the projection non-linear and precludes using a linear Kalman filter. The filter linearizes the projection equation around the predicted camera system pose to form the Jacobian used in the EKF equations. 3D feature estimates are kept in memory only so long as the feature is tracked by the 2D tracker. This limits the total memory usage of the filter, enabling tracking over large areas. It also means that the filter cannot perform loop completion which is a common limitation of most structure from motion systems when processing long video sequences covering large areas.

One of the main drawbacks of using the EKF is that as the number of tracked salient features increases, the storage space required to store the filter's covariance matrix increases in a quadratic fashion because the matrix stores all of the feature covariances and

their cross-covariances. The filter's update cycle complexity is $O(n^3)$ where n is the number of features. This makes real time operation on large sets of features problematic. We avoid this performance bottleneck by taking advantage of the statistical independence of the salient features. So long as the features are stationary with respect to the world coordinate frame, their cross-covariance terms are zero in the filter's covariance matrix. This yields a large, sparse matrix. The structure of this matrix could be exploited to speed up the inversion step, which is part of the Kalman filter.

Another approach is to process the feature measurements, which are taken at the same time, sequentially. This approach to processing in the Kalman filter is described in [2]. The filter update cycle starts by predicting the camera position and covariance at the next time step. Then the filter processes each of the 2D feature measurements in sequential fashion. In each sequential update a subset of the total state comprised of the camera system state and a randomly selected 3D feature estimate is generated and processed to update the filter's state and covariance estimate as well as the position and covariance of the 3D feature. Each 2D feature measurement that is processed reduces the uncertainty of the camera pose a certain amount as well as the uncertainty of the corresponding 3D feature. When processing the features sequentially, features that are processed earlier tend to have a greater influence on the camera pose estimate but only because they cause a correction to the state which later measurements support. So long as features are processed in random order, over time sequential processing can be shown to behave similarly to processing all features at once in a single update cycle [21].

One advantage of sequential processing is that it allows simple outlier detection and rejection. Outliers are detected based on the difference between the estimated 3D point's projection and its corresponding measurement in the current frame. This error is the filter's residual which is an integral component of the Kalman filter. Outliers are not allowed to influence the camera system state and covariance and are removed from the total filter state.

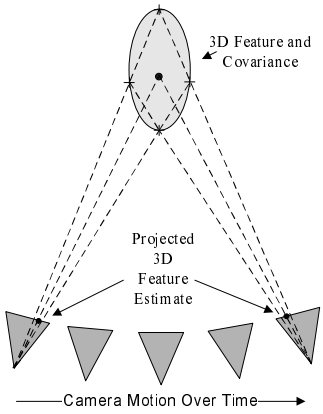


Figure 2: Initial covariance sampling

The initialization of 3D features and their covariances is an important part of the filter design. The filter should strive to initialize 3D estimates for 2D tracks only for inliers. Feature initialization in the leading filter is done by triangulation across a minimum base-

line. In addition, the angle between the rays to the feature is measured and a threshold is applied to this angle. In this our filter implementation we chose a threshold of 10^o. This prevents features at infinity from being processed by the filter. (Features at infinity give information about camera rotation but no information about translation and have very large uncertainties, which could cause numerical problems.) Further, the triangulated 3D point is projected into each of the cameras that it has been tracked in 2D so far in the sequence. Only projected features that are within 1 pixel of their corresponding measurements are passed into the filter. This threshold includes both expected measurement error in the KLT tracker, as well as error in the filter's camera pose prediction.

Initial 3D feature covariances are determined by generating a sampled probability distribution in 3D. This is done by intersecting the perturbed rays corresponding to the projected 3D feature estimate in the first frame it is tracked in and the current frame. Each ray is perturbed by the expected amount of measurement noise in eight directions around the projected 3D point in the horizontal, vertical and diagonal directions in the two frames' image spaces. A Gaussian distribution is then fit to this set of samples. A simplified example of this sampling process, sampling only in the horizontal direction, is shown in figure 2.

3.2 Two-Stage Extended Kalman Filter Pipeline

In the previous section we described the operation of a single structure from motion process performed by an extended Kalman filter. Our novel approach combines two single filters staggered in time and operating in parallel to improve SfM accuracy. The filter leading in time selects the best set of inliers and initializes their estimated 3D coordinates and uncertainties. Inliers are then passed to the filter following in time, which performs SfM only on the inliers equiped with reliable initial estimates and covariances, improving the SfM accuracy in the following filter.

The leading filter operates on the current frame in the video sequence and selects the best 2D feature tracks, passing initial estimates of the 3D feature locations and feature covariances to the following filter. The following filter operates a fixed number of frames behind the leading filter in the video sequence. Because the following filter receives 3D feature estimates and covariances from the leading filter, it is able to track 2D features from the frame where the 2D track begins and does not have to wait to triangulate the feature or convert feature tracks from a ray/depth/camera center formulation to full 3D formulation, which is done in recent Kalman filter based SLAM implementations [4]. This increases the overall number of good features that are tracked in the following filter each frame.

This two-stage architecture allows a simple and effective form of measurement selection. Features are selected to be passed to the following filter if they are triangulated and then tracked in 3D for a fixed number of frames. Outlier 2D tracks may occasionally be triangulated and added to the leading filter state. However, it is unlikely that these outliers will continue for more than a couple of frames in the leading filter without being rejected as outliers based on their higher reprojection errors due to their inconsistency with the camera motion. By only passing back 3D features that last multiple frames in the leading filter, the following filter processes only features which are consistent with the camera motion. This makes the following filter's camera pose and scene structure estimates more accurate, as demonstrated by reduced reprojection errors. The feature initialization pro-

cess is shown in figure 3. In that figure dashed lines represent measurements of the 3D feature in a given camera.

Using this architecture the leading and following filters states are not bound together and so the state estimates could drift apart over time. Still both of the cameras' relative motions should be approximately the same over a short time span. The 3D feature locations estimated by the leading filter, which estimates the 3D features in a world coordinate frame, can be passed to the following filter by passing the feature's position relative to the leading camera pose which corresponds to the following filter's current state. In this way the two filters' states are coupled together through the initial 3D feature estimates.

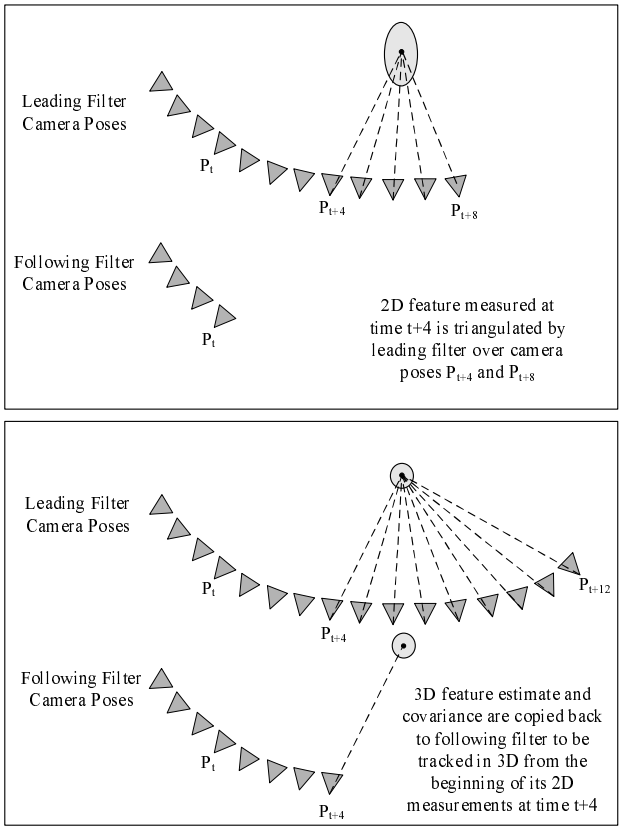


Figure 3: Initialization of a feature in the leading filter and passing the feature back to the following filter

Our pipelined estimation approach is considerably more efficient that a naive lookahead filter implementation. A naive implementation when estimating the camera pose and scene structure at frame i would process all of the measurements for frames i to $i+w$, where w is the number of frames looked ahead, to find the inlier correspondences to integrate into the final estimate at frame i, repeating this process of looking ahead w frames and then taking a single step at each frame. This would yield an overall compu-

tational complexity of $O(wh)$ where w is the number of frames looked ahead and h is the cost of performing one complete SfM estimation on the video sequence. In contrast, our two filter approach is able to determine which correspondences are reliable inliers with a computation cost of only $O(h)$.

4 Results

To demonstrate the improved performance of our two-stage approach we ran the tracking system over ten seconds of video. The video was collected using a camera with known intrinsic calibration and a field of view of approximately 40^ox30^o, frame rate of 30 frames per second and resolution of 1024x768 pixels. The camera was rigidly coupled to an inertial navigation system which was used to initialize the Kalman filter's velocity and rotational velocity estimates. This was necessary because the Kalman filter formulation we use is tuned for a particular scale of motion and so the initial scaled translation and rotation rates must be known. One could just as easily initialize the filter with a fiducial of known size. Recently, Civera [5] has devised a parametrization of structure from motion estimation for the Kalman filter that does not require scale initialization and that could be used in our two-stage architecture to mitigate this limitation.

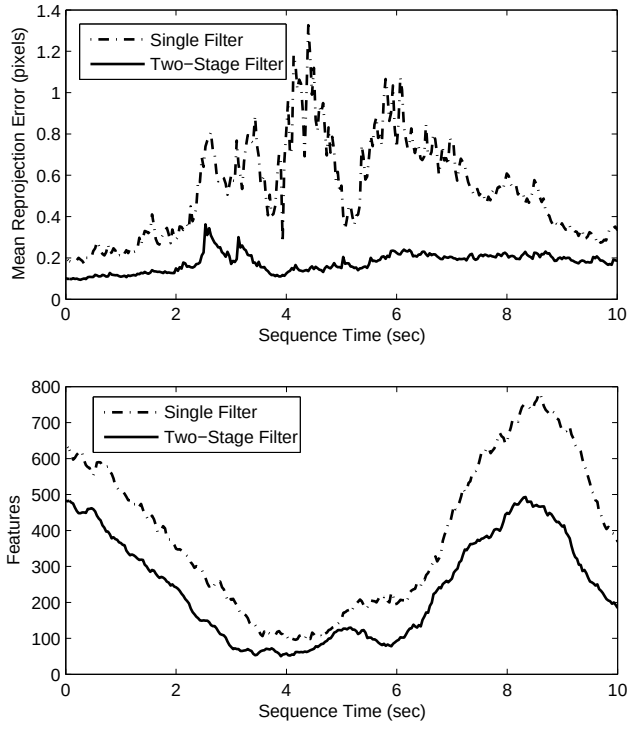


Figure 4: Above: Comparison of mean reprojection error per frame Below: Features included in mean reprojection error per frame

Figure 4 shows the improvement in reprojection errors by selecting measurements and initializing 3D feature estimates using the lookahead filter. The graph shows the mean reprojection errors of all 3D features tracked in each frame, projected into every frame in which they are tracked in 2D. One can clearly see that selecting only those features that are tracked in 3D in the leading filter for 4 or more frames and tracking only those features in the following filter significantly improves the tracking performance of the following filter. No additional non-linear optimization is performed on these results. Figure 4 shows the number of features tracked in 3D using only a single filter which is identical to the leading filter vs. using the two-stage filter architecture. This demonstrates the ability of the two-stage pipelined filter system to select a superior subset of the tracks generated by a single filter system.

5 Conclusion

In this paper we have introduced a measurement selection and initialization approach utilizing a two-stage filter architecture to determine the best set of features and initialize their estimates and uncertainties. These features have lower reprojection errors when processed, allowing for more accurate structure from motion estimation than approaches that attempt to estimate structure from motion in the most recently captured frame with no delay. Pipelined estimation is applicable to many types of robust estimation systems including Kalman and particle filters and is applicable to any system of potentially unreliable sensors, where a reliable set of sensors must be selected and a small delay in estimating the state is acceptable.

Future work on pipelined estimation may involve selecting an optimal set of good features to process (minimal computational cost to process with maximal camera state information) in the following filter which gives a reliable camera pose estimate while minimizing the computational cost of operating multiple filters, allowing for real time filter operation with high accuracy. Additionally, in the current architecture it is possible for the leading filter's scale to drift away from the following filter's over time. Addressing this potential weekness by correcting the leading filter's state using the following filter's more reliable estimates (feeding them forward) would make the system more robust to scale or other drift between the filters.

6 Acknowledgements

We would like to acknowledge the David and Lucille Packard Foundation Fellowship for funding this work.

References

[1] Ali Azarbayejani and Alex P. Pentland. Recursive estimation of motion, structure, and focal length. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(6):562–575, 1995.

[2] Robert G. Brown and Patrick Y. C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering, 3rd Edition*. John Wiley and Sons, New York, 1997.

[3] A. Chiuso, P. Favaro, H. Jin, and S. Soatto. Structure from motion causally integrated over time. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):523–535, 2002.

[4] Javier Civera, Andrew J. Davison, and J. M. M. Montiel. Inverse depth to depth conversion for monocular slam. In *ICRA*.

[5] Javier Civera, Andrew J. Davison, and J. M. M. Montiel. Dimensionless monocular slam. In *Iberian Conference on Pattern Recognition and Image Analysis*, 2007.

[6] A. Davison. Real-time simultaneous localisation and mapping with a single camera. 2003.

[7] Ethan Eade and Tom Drummond. Scalable monocular slam. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 469–476, Washington, DC, USA, 2006. IEEE Computer Society.

[8] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[9] B.D. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *Int. Joint Conf. on Artificial Intelligence*, pages 674–679, 1981.

[10] P. Meer. *Robust techniques for computer vision. Emerging Topics in Computer Vision, G. Medioni and S. B. Kang (Eds.)*, pages 107–190. Prentice Hall, 2004.

[11] Peter Meer, Doron Mintz, Azriel Rosenfeld, and Dong Yoon Kim. Robust regression methods for computer vision: a review. *Int. J. Comput. Vision*, 6(1):59–70, 1991.

[12] J.Ṽ. Miller and C.Ṽ. Stewart. Muse: robust surface fitting using unbiased scale estimates. In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, 1996 IEEE Computer Society Conference on*, pages 300–306, 1996.

[13] H. E. Rauch. Solutions to the linear smoothing problem. *IEEE Transactions on Automatic Control*, 82(4):371–372, 1963.

[14] H. E. Rauch, F. Tung, and C. T. Striebel. On the maximum likelihood estimates for linear dynamic systems. Technical Report 6-90-63-62, Lockheed Missiles and Space Company, Palo Alto, CA, June 1963.

[15] H.E. Rauch, F. Tung, and C.T. Striebel. Maximum likelihood estimates of linear dynamic systems. *AIAA Journal (American Institute of Aeronautics and Astronautics)*, 3(8):1445–1450, August 1965.

[16] P. J. Rousseeuw and A. M. Leroy. *Robust regression and outlier detection*. John Wiley & Sons, Inc., New York, NY, USA, 1987.

[17] J. Shi and C. Tomasi. Good Features to Track. In *Int. Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[18] Charles V. Stewart. Minpran: A new robust estimator for computer vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(10):925–938, 1995.

[19] Chi-Keung Tang, Gerard G. Medioni, and Mi-Suen Lee. Epipolar geometry estimation by tensor voting in 8d. In *ICCV (1)*, pages 502–509, 1999.

[20] Philip H. S. Torr and Andrew Zisserman. Robust computation and parametrization of multiple view relations. In *ICCV*, pages 727–732, 1998.

[21] Greg Welch and Gary Bishop. Scaat: Incremental tracking with incomplete information. In Turner Whitted, editor, *Computer Graphics*, Annual Conference on Computer Graphics and Interactive Techniques, pages 333–344. ACM Press, Addison-Wesley, Los Angeles, CA, USA (August 3-8), siggraph 97 conference proceedings edition, 1997.

Time Varying Volumetric Scene Reconstruction Using Scene Flow

Timothy Smith, David Redmill, Nishan Canagarajah, David Bull
Department of Electrical and Electronic Engineering,
University of Bristol, BS8 1UB, UK
timothy.smith@bristol.ac.uk

Abstract

Traditional volumetric scene reconstruction algorithms involve the evaluation of many millions of voxels which is highly time consuming. This paper presents an efficient algorithm based of future frame prediction that can dramatically reduce the number of voxels to be evaluated in time varying scenes. The new prediction method, combining scene flow and morphological dilations, is evaluated against a simple model dilation method. Results show the proposed method outperforms a simple dilation method and has the potential to improve the efficiency of volumetric scene reconstruction algorithms while retaining quality given accurate optical flows.

1 Introduction

Volumetric scene representations use a compact three dimensional grid to record colour and occupancy information at discrete points in space. Images taken from multiple calibrated cameras can be used to populate this voxel grid and many algorithms have been proposed [11, 7, 3, 16]. These algorithms have mainly been targeted at static scenes with their extension to video consisting of frame by frame processing. This results in useful temporal information being ignored which could otherwise aid the reconstruction process. This paper proposes a simple method of incorporating the observed optical flow from each camera into the reconstruction process with a view to dramatically reducing the number of voxels evaluated at each time frame. From per camera dense optical flows the per voxel scene flow is calculated and used to produce a volumetric frame prediction which is then dilated. This predicted model is then used to guide the voxel estimation of the next frame. An overview of voxel reconstruction algorithms is given in Section 2 with details of the proposed algorithm in Section 3. Section 4 shows results using the proposed technique and conclusions are drawn in Section 5.

2 Review of Volumetric Reconstruction

This section provides details of some of the main volumetric reconstruction algorithms relevant to this paper. A good overview of scene reconstruction techniques can be found in [12].

The voxel colouring algorithm was introduced in [11] as a method of constructing a set of voxels with associated colours from a set of calibrated colour images. If deemed

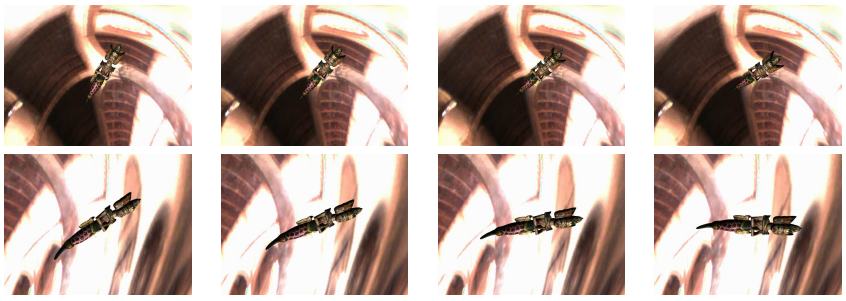


Figure 1: Original synthetic frames from 2 cameras. Left to right: Frame 1, 7, 13 and 19.

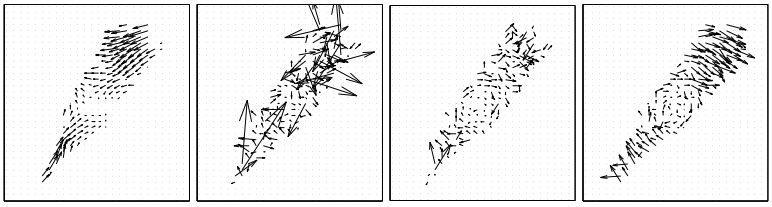
to be colour consistent across the images, a voxel is marked as occupied and assigned an average colour otherwise it is marked as transparent. Voxel occlusions are handled by restricting camera placement to satisfy the *ordinal visibility constraint*.

Space carving is described in [7] as a generalization of voxel colouring. In [7] it is shown that by starting with an overcomplete voxel representation of a scene then removing non-photoconsistent voxels, a *photo hull* can be produced. The space carving algorithm does not impose restrictions on camera placement by using a multi-sweep algorithm. Generalized voxel colouring [3] and multi-hypothesis reconstruction [4] attempt to solve the voxel visibility problem using slightly different techniques. In [3] each voxel is carved based on its simultaneous photoconsistency in all camera views in which it is visible while in [4] each voxel is assigned a number of colour hypotheses which are gradually removed if inconsistent. While space carving-type algorithms [4, 3, 7] are more general than voxel colouring [11] they still suffer from many of the same reconstruction artifacts, notably fattened reconstructions where surfaces bulge out towards the cameras.

Basing the voxel occupancy decision on a local threshold generally results in a non-optimal solution being reached. Rather, a globally optimum solution should be found. Vogiatzis et al. [16] extract a photoconsistent object surface using a minimum cut solution of a weighted graph representation of a photoconsistency cost function. In [6] this graph cut algorithm is enhanced by allowing adjacent voxels to contribute to the photo consistency function and implicitly providing a smoothing term. This technique is currently limited to closed, watertight objects.

A number of techniques to bring voxel colouring closer to real-time performance are suggested in [10]. Using temporal coherence to speed up voxel colouring of dynamic scenes is suggested with a simple extension that takes the previous frame in a sequence, dilates the occupied voxel set from that frame and then uses this set as the search space of the current frame. While a speed up of around two is shown, the method is unsuitable for fast moving scenes as no explicit motion parameters are calculated.

Some work has also been done with regard to modelling moving scenes using voxels. In [15] a six dimensional voxel representation of a scene is proposed where voxels are carved if they are inconsistent with images from two time instants or inconsistent with the flow between the two times. This method also recovers the scene flow between frames. The scene flow [14] can also be derived from per camera 2D optical flows which in [13] is applied to interpolating between two already carved scene frames. Rather than using two known scenes at consecutive times, this paper takes scene flow and uses it to project a known scene forward in time to guide the estimation of the following scene frame.



(a) H264 (b) Horn-Schunck (c) non-iterative L-K (d) hierarchical L-K

Figure 2: Optical flows for synthetic sequence, frame 1.

3 Methodology

A volumetric reconstruction method for moving scenes is proposed which consists of three steps: voxel occupancy estimation, voxel scene flow estimation and voxel occupancy prediction. A volumetric representation of the first frame in the sequence is estimated by evaluating all voxels in the scene volume. This gives an occupancy and colour for each voxel. The scene flow is then calculated from per camera optical flows and, combined with a dilation step, used to predict the occupancy in the next frame. When subsequent frames are processed only voxels which have been predicted as occupied from the previous frame are evaluated. More details are given below.

3.1 Voxel Occupancy Estimation

The voxel occupancy step can be any algorithm that takes calibrated input images and produces a colour and occupancy for each voxel in the scene, such as voxel colouring [11], voxel carving [3, 4, 7] or volumetric graph-cuts [16]. In the rest of this paper, voxel colouring is used for the voxel occupancy estimation due to its simplicity.

3.2 Optical Flow

Video sequences are often analyzed using optical flow. A very brief overview of the optical flow estimation techniques used in this paper is given below. More details can be found in the original papers [5, 8, 2] and a performance analysis of techniques in [1]. If optical flow is thought of as a simple translation, $\mathbf{v} = (\frac{\partial u}{\partial t}, \frac{\partial v}{\partial t})^T$, and intensity is assumed to be conserved then the gradient constraint equation may be written

$$\nabla I(\mathbf{x},t) \cdot \mathbf{v} + I_t(\mathbf{x},t) = 0 \tag{1}$$

where $I_t(\mathbf{x},t) = \frac{\partial I(\mathbf{x},t)}{\partial t}$. A second constraint [1] must be used to solve this equation.

Horn and Schunck [5] use a global smoothness term to provide this extra constraint and seek to iteratively minimize

$$E = \int_D (\nabla I \cdot \mathbf{v} + I_t)^2 + \lambda^2 (\|\nabla u\|^2 + \|\nabla v\|^2) d\mathbf{x} \tag{2}$$

Lucas and Kanade [8] assume optical flows are constant in a small neighbourhood and seek to minimize the error function

$$E = \sum_{\mathbf{x} \in \Omega} W^2(\mathbf{x}) \left[\nabla I(\mathbf{x},t) \cdot \mathbf{v} + I_t(\mathbf{x},t)\right]^2 \tag{3}$$

where $W^2(\mathbf{x})$ is a window function with decreasing weights from its centre. An iterative scheme may be applied where the source image is warped towards the target image after each minimization step using the current estimate of the optical flow. The estimated optical flow between the target image and warped source image is then computed and added into the overall optical flow.

Both of these gradient based techniques assume small (less than a pixel) optical flows. To combat this limitation a hierarchical scheme may be used [2]. A Gaussian pyramid is constructed from two temporally adjacent original images and the optical flow estimation run on the lowest resolution images in each of the pyramids. This flow information is then propagated to the next highest resolution to form the starting point for that resolution. This allows a straightforward integration into the iterative Lucas-Kanade algorithm.

Optical flow may also be solved directly using block matching techniques which are commonly performed by searching in $\mathbf{v}$ to minimize the sum of squared differences between the source and target block. The reader is referred to [1] for a fuller discussion. Such block matching algorithms are often found in motion based video compression schemes such as MPEG-2 and H264 [9] with the motion vectors embedded into the compressed video stream.

3.3 Scene Flow

In the scene flow estimation step the motion of each voxel is represented using scene flow as introduced in [14] as a 3D extension of optical flow in 2D. Let $\mathbf{x}(t) = (x, y, z)$ be the position of a 3D scene point (voxel centre) at time t and $\mathbf{u}_n(t) = (u_n, v_n)$ be its projection in image I_n then

$$\frac{d\mathbf{u}_n}{dt} = \frac{\partial \mathbf{u}_n}{\partial \mathbf{x}} \frac{d\mathbf{x}}{dt} \tag{4}$$

where $\frac{d\mathbf{u}_n}{dt}$ is the optical flow in image n and $\frac{d\mathbf{x}}{dt}$ is the instantaneous scene flow. A system of equations $\mathbf{B}\frac{d\mathbf{x}_j}{dt} = \mathbf{A}$ can be set up with $N \geq 2$ cameras where

$$\mathbf{B} = \begin{bmatrix} \frac{\partial u_1}{\partial x} & \frac{\partial u_1}{\partial y} & \frac{\partial u_1}{\partial z} \\ \frac{\partial v_1}{\partial x} & \frac{\partial v_1}{\partial y} & \frac{\partial v_1}{\partial z} \\ \vdots & \vdots & \vdots \\ \frac{\partial u_N}{\partial x} & \frac{\partial u_N}{\partial y} & \frac{\partial u_N}{\partial z} \\ \frac{\partial v_N}{\partial x} & \frac{\partial v_N}{\partial y} & \frac{\partial v_N}{\partial z} \end{bmatrix}, \mathbf{A} = \begin{bmatrix} \frac{\partial u_1}{\partial t} \\ \frac{\partial v_1}{\partial t} \\ \vdots \\ \frac{\partial u_N}{\partial t} \\ \frac{\partial v_N}{\partial t} \end{bmatrix} \tag{5}$$

By taking the singular value decomposition of $\mathbf{B}$ such that $\mathbf{B} = \mathbf{U}.\mathbf{w}.\mathbf{V}^T$, a solution can be found for the j^{th} voxel as $\frac{d\mathbf{x_j}}{dt} = \mathbf{V}.diag(\frac{1}{w_i}).\mathbf{U}^T.\mathbf{A}$. This solution minimizes the squared error between the reprojected scene flow and the optical flow in each camera image. If more cameras are used, a more robust scene flow estimation can be calculated. $\mathbf{B}$ is calculated from the camera projection matrix at $\mathbf{x}(t)$. The scene flow is calculated for each voxel marked as occupied to create a 3D volumetric model that includes per voxel motion.

Voxel evaluation method	Average PSNR (dB)
All voxels	14.0272
4 dilations only	13.7315
Hierarchical Lucas-Kande + 3 dilations	13.6506
Lucas-Kanade + 3 dilations	13.5858
H264 + 3 dilations	13.2917
Horn-Schunck + 3 dilations	13.2564
3 dilations only	12.3454

Table 1: Average PSNR for different frame prediction methods for synthetic sequence.

3.4 Voxel Occupancy Predication

The next step is to predict the next scene frame based on the current frame. To do this each voxel in the current scene frame is moved based on the scene flow vector assigned to it with voxels which would be moved out of the volume being clamped to lie on the edge of the volume. Using scene flow on its own is not sufficient for successful prediction therefore this paper proposes that the predicted model is then expanded using a 3D version of the morphological dilation operator which dilates based on each voxel's six-face-connected binary occupancies. There are three motivations to doing this:

- Each predicted voxel is forced to lie on integer voxel coordinates whereas in reality it lies between integer voxels and has an influence on the surrounding voxels.
- An unknown error is associated with each scene flow vector leading to voxels possibly being moved incorrectly.
- The forward flowed voxel model may have holes which would affect subsequent reconstructions as voxels are only removed, never added, during voxel carving.

The number of dilations is found empirically based on the granularity of the voxel model compared with the input images and the error associated with the optical flow field. With a fine voxel model, a number of dilations may correspond to a single pixel change in the input images meaning more dilations are needed.

4 Results and Discussion

For the evaluation of the proposed technique a synthetic 20 frame sequence[1] (Figure 1) and 20 frame natural sequence[2] (Figure 7a) from 8 fully calibrated cameras were used.

In the synthetic sequence the highly textured figure rotates with its extremities moving at 5 to 8 pixels per frame while its centre of rotation remains fixed. Optical flows were recovered for every frame in every camera using three methods: Horn-Schunck, non-iterative Lucas-Kanade (as in [1]) and hierarchical iterative Lucas-Kanade. In addition, the block motion vectors from an H264 encoding of each camera sequence found using the exhaustive search strategy [9] were extracted and upsampled to produce an H264 optical flow. For the spatial image derivatives a 5 point central difference kernel was used

[1] Original 3D model Copyright ©Andrew Kator, http://www.katorlegaz.com/

[2] Dataset from Interactive Visual Media Group, Microsoft Research [17]

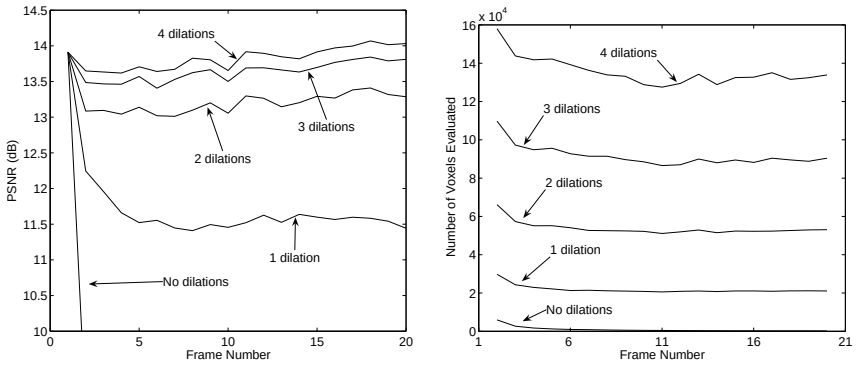


Figure 3: PSNR and voxel count using proposed technique on synthetic sequence with hierarchical Lucas-Kanade optical flow and a varying number of dilations.

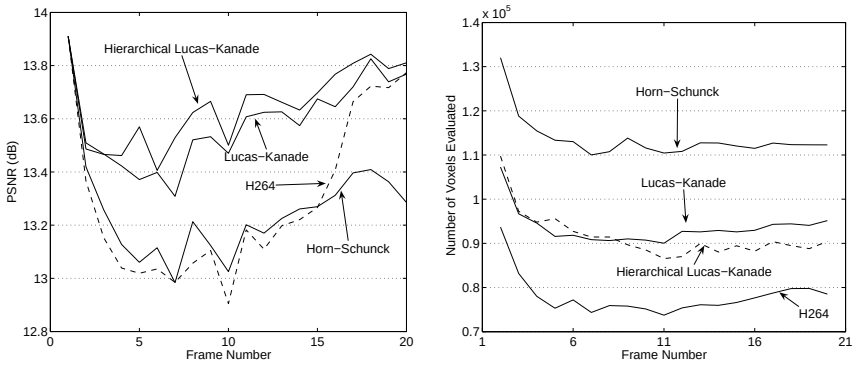


Figure 4: PSNR and voxel count using proposed technique on synthetic sequence with different optical flow estimation algorithms and 3 dilations.

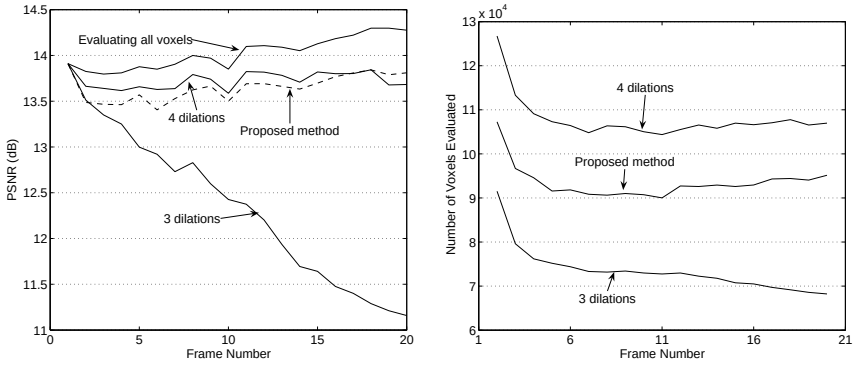
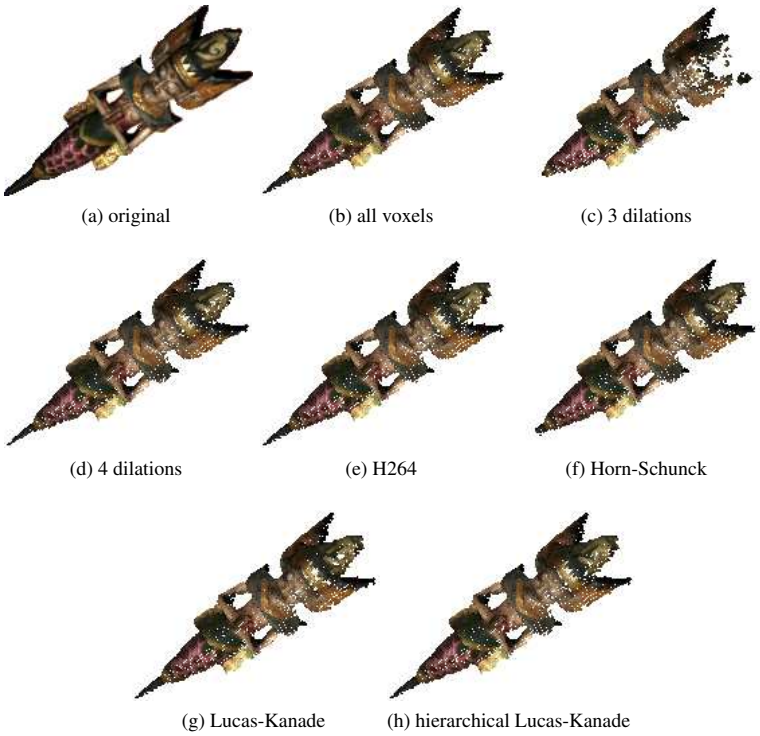


Figure 5: PSNR and voxel count comparing only using dilations, evaluating all voxels and using proposed technique (Lucas-Kanade plus 3 dilations) for synthetic sequence.



(a) original (b) all voxels (c) 3 dilations

(d) 4 dilations (e) H264 (f) Horn-Schunck

(g) Lucas-Kanade (h) hierarchical Lucas-Kanade

Figure 6: Frame 19 reconstructions for synthetic sequence.

(as in [1]) while the temporal derivative was calculated from a simple frame difference. The original images were not pre-smoothed. Example flows for each method for frame 1 are shown in Figure 2. As expected the hierarchical iterative Lucas-Kanade (Figure 2d) retrieves the most accurate optical flows with large motions correctly recovered. All the optical flow algorithms struggled to obtain accurate flows for the 'tail' of the figure which is near homogeneous in colour and in many images is the same colour as the background.

These four sets of optical flows were then used in the proposed reconstruction algorithm using 3 dilations, chosen to give a balance between scene reconstruction quality and voxel count. Figure 3 shows the sensitivity of the proposed algorithm to varying the number of dilations. To assess the quality of the estimated model each frame was reconstructed for each camera and a peak signal-to-noise ratio (PSNR) calculated between the reconstruction and the original camera frame image. This PSNR was calculated over the actual image area of the original 3D model based on a segmentation of the background and foreground generated when the original scene was being rendered. The mean frame PSNR over all cameras and the number of voxels evaluated at each frame, excluding frame 1, are shown in Figure 4. Frame 1 has $256^3 \approx 10^7$ voxels evaluated.

It is clear that the best performance is achieved using the hierarchical iterative Lucas-Kanade optical flows. Notably there is a sharp drop in PSNR after the first frame indicating the shift away from evaluating all voxels to evaluating only predicted voxels.

(a) original (b) all voxels evaluated (c) proposed method

Figure 7: Frame 8 reconstructions for natural sequence

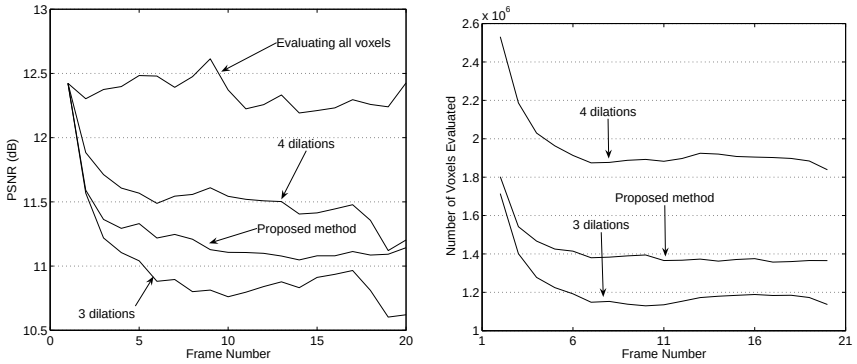


Figure 8: PSNR and voxel count comparing only using dilations, evaluating all voxels and using proposed technique (Lucas-Kanade plus 3 dilations) for natural sequence.

Interestingly, the H264 PSNR continues to drop until frame 10 whereupon it starts to rise again. This can be explained by looking at the optical flows estimated for each frame from the block motion vectors and noting that for the first 10 frames the direction of movement at the head of the figure is incorrect[3]. In the later frames this block is estimated correctly. Despite causing the most voxels to be evaluated (Figure 4), the Horn-Schunck optical flows produce the worst reconstruction results due to the badly estimated optical flow in areas with large pixel displacements (Figure 2b).

Taking the best performing optical flow (hierarchical Lucas-Kanade) allows a comparison to be made to a simple dilation method [10] as well as to evaluating all voxels in the scene. As the motion predicted model based on scene flow is dilated three times before being used, it is important to establish that the same effect could not be achieved simply using 3 dilations alone. Figure 5 clearly shows that 3 dilations is insufficient to track the object whereas incorporating optical flow brings the PSNR back up to a stable state. Related to this PSNR drop is the drop in the number of voxels (Figure 5) being evaluated meaning that voxels are continuously being lost from the constant volume object. Increasing the number of dilations to 4 brings the PSNR to a value similar to that obtained using scene flow at the expense of an increase in the number of voxels evaluated. The dilations must expand the previous model enough to capture the largest motion in the scene meaning a scene with large motions needs more dilations to capture the voxels associated

[3]Incorrect in terms of optical flow. The H264 estimated motion is actually that which minimizes the coding cost and is most probably correct in this sense.

with these motions. Using scene flow to guide the prediction allows large motions to be present in the scene and still keep the number of voxels evaluated to a minimum.

Evaluating all voxels in a scene produces the best quality reconstruction but using the proposed scene flow guided prediction model as a hypothesis for the next scene frame leads to only a small drop in quality with a dramatic reduction in voxel evaluations performed. As reconstruction time is directly proportional to the number of voxels evaluated and the optical flow calculations are relatively fast, a significant processing time decrease is achieved. A summary of the overall average frame PSNRs is shown in Table 1 while Figure 6 shows the synthesis results for frame 19 from a single camera. The low overall PSNR could be improved by using a more sophisticated voxel occupancy algorithm [16].

Results for the natural sequence are shown if Figures 7 and 8 with PSNR evaluated over the entire image. The extremities of the dancer move upto 80 pixels between frames leading to very poor optical flow estimation for these areas which leads to the observable poor synthesis in these regions (Figure 7c). Even so, including optical flow still produces higher quality syntheses than using only 3 dilations and is comparable to the synthesis quality obtained when evaluating all voxels.

5 Conclusions and Future Work

Volumetric scene reconstruction algorithms usually focus on static scenes and do not take into account temporal information when reconstructions are performed on moving scenes. To address this weakness, this paper has suggested using a combination of scene flow and morphological dilations applied to a standard voxel colouring algorithm. A number of optical flow algorithms [5, 8, 2] have been used to obtain dense scene flow which has then been applied to the prediction of future scene frames. Basing voxel occupancy estimation on the predicted occupancy allows a dramatic decrease in the number of voxels which need to be evaluated leading to a substantial computational speed gain with only a small decrease in reconstruction quality. The proposed technique also improves on previous model dilation techniques [10].

At present, evaluating all voxels in the scene for photoconsistency produces the highest quality reconstructions. In the future, techniques for increasing the reconstruction quality based on scene flow will be explored, such as dynamically varying the number of dilations based on optical flow confidences. The same scene flow based algorithm will also be integrated with more advanced voxel occupancy estimation algorithms.

References

[1] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of Optical Flow Techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.

[2] James R. Bergen, P. Anandan, Keith J. Hanna, and Rajesh Hingorani. Hierarchical Model-Based Motion Estimation. In *Proceedings of the European Conference on Computer Vision*, volume LNCS 588, pages 237–252, 1992.

[3] W. Bruce Culbertson, Thomas Malzbender, and Gregory G. Slabaugh. Generalized Voxel Coloring. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 100–115, 2000.

[4] P. Eisert, E. Steinbach, and B. Girod. Multi-Hypothesis, Volumetric Reconstruction of 3-D Objects from Multiple Calibrated Camera Views. In *International Conference on Acoustics Speech and Signal Processing*, pages 3509–3512, March 1999.

[5] Berthold K. P. Horn and Brian G. Schunck. Determining Optical Flow. *Artificial Intelligence*, 17:185–203, 1981.

[6] Alexander Hornung and Leif Kobbelt. Robust and Efficient Photo-Consistency Estimation for Volumetric 3D Reconstruction. In *Proceedings of the European Conference on Computer Vision*, volume LNCS 3952, pages 179–190, May 2006.

[7] Kiriakos N. Kutulakos and Steven M. Seitz. A Theory of Shape by Space Carving. *International Journal of Computer Vision*, 38(3):199–218, 2000.

[8] Bruce D. Lucas and Takeo Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)*, pages 674–679, April 1981.

[9] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi. Video coding with H.264/AVC: Tools, Performance, and Complexity. *IEEE Circuits and Systems Magazine*, 4(1):7–28, 2004.

[10] C. Prock and A. Dyer. Towards Real-Time Voxel Coloring. In *DARPA Image Understanding Workshop*, pages 315–321, 1998.

[11] S. M. Seitz and C. R. Dyer. Photorealistic Scene Reconstruction by Voxel Coloring. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1067–1073, June 1997.

[12] G. Slabaugh, B. Culbertson, T. Malzbender, and R. Schafer. A Survey of Methods for Volumetric Scene Reconstruction from Photographs. In *International Workshop on Volume Graphics*, pages 81–100, June 2001.

[13] Sundar Vedula, Simon Baker, and Takeo Kanade. Image-Based Spatio-Temporal Modeling and View Interpolation of Dynamic Events. *ACM Transactions on Graphics*, 24(2):240–261, April 2005.

[14] Sundar Vedula, Simon Baker, Peter Rander, Robert Collins, and Takeo Kanade. Three-Dimensional Scene Flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):475–480, March 2005.

[15] Sundar Vedula, Simon Baker, Steven Seitz, and Takeo Kanade. Shape and Motion Carving in 6D. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 592–598, June 2000.

[16] G. Vogiatzis, P.H.S. Torr, and R. Cipolla. Multi-View Stereo via Volumetric Graph-Cuts. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 391–398, June 2005.

[17] C. Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High-quality video view interpolation using a layered representation. *ACM Transactions on Graphics*, 23(3):600–608, 2004.

Human Pose Extraction from Monocular Videos using Constrained Non-Rigid Factorization

Appu Shaji
Dept. of Computer Science and Engineering
IIT Bombay
Powai, Mumbai 400 076, India
appu@cse.iitb.ac.in

Behjat Siddiquie
Dept. of Computer Science
University of Maryland
College Park, MD 20742, USA
behjat@cs.umd.edu

Sharat Chandran
Dept. of Computer Science and Engineering
IIT Bombay
Powai, Mumbai 400 076, India
sharat@cse.iitb.ac.in

David Suter
Dept. of Electrical and Computer Systems Engineering
Monash University
Clayton 3800, Victoria, Australia
d.suter@eng.monash.edu.au

Abstract

We focus on the problem of automatically extracting the 3D configuration of human poses from 2D image features tracked over a finite interval of time . This problem is highly non-linear in nature and confounds standard regression techniques. Our approach effectively marries a non-rigid factorization algorithm with prior learned statistical models from archival motion capture database. We show that a stand alone non-rigid factorization algorithm is highly unsuitable for this problem. However, when coupled with the learned statistical model in the form of a constrained non-linear programming method, it yields a substantially better solution.

1 Introduction

Given a monocular video which features a single human in motion, our goal in this work is to reconstruct the 3D configuration (seen from an arbitrary choice of a world coordinate system). We assume that we have as input anatomically well-defined landmark points (such as major joints) recorded from an orthographic or weak-perspective camera. Our emphasis is not in feature tracking, but rather on recovering the lost depth during image formation from noisy and possibly incomplete data.

Human motion comprises of an enormous amount of inherent subtlety and variability. Consequently the problem of inferring 3D pose from 2D image sequences is highly non-linear in nature and confounds standard regression techniques. Besides, even if we have a good knowledge about the projection matrix of the camera, for any single input observation of a human pose in 2D, there are possibly multiple valid body configurations. Correlate this with our lack of judgment when we see the Necker cube. From a numerical point of view, estimating 3D structure and motion from image sequences is a higher order (quartic) non-linear optimization problem (§Eq. 5), prone to local minima. These local minima are intrinsic to the problem (termed as true illusions [1]).

Previous Work: A variety of statistical as well as deterministic methods have been developed for extracting pose from single view image sequences. We can define a gross dichotomy on the class of approaches: Ones that concentrate on learning a mapping from silhouette feature space to 3D pose [2], and others that try to map feature points, usually localized to anatomically meaningful landmark points such as elbows position, limb end-point position etc. to 3D poses [3, 4]. Our approach falls in the second category. For a curious reader, we suggest [5] which catalogs most of the important works on 3D human tracking.

The solution approach in all of the above cases sans [4] is formulated as an (approximate) probabilistic inference problem. Given an observation, they try to pick a pose from a prior distribution which best fits the current likelihood. Though this is an extremely powerful tool, we note that the methods do not explicitly address geometric properties or algebraic details of the data. Rather, the methods rely on these details being captured during the training stage and appear as latent parameters. In essence, this transfers too much importance to the training stage.

An alternative less explored, is to borrow techniques from structure from motion (SfM) and couple them with prior statistical knowledge. SfM [6] techniques are able to produce highly accurate solution when the object is rigid, and is widely regarded as one of biggest success story of computer vision. But, extending SfM to non-rigid scenario has turned out to be quite tricky. One popular flavor of SfM algorithm is the Factorization algorithm [7–10].

In this work, we use a variant of recently proposed [10] non-rigid factorization method (NRF, hereafter) for performing SfM.

Methodology: Factorization methods attempt to capture the implicit geometric invariants present in a wide temporal window of input data. (An example invariant might be that two feature points from a single rigid body should have similar motion trajectories. These invariants uncover themselves as reduced rank constraints [7, 8, 10] on the data observation matrix consisting of stacked (x,y) points. This matrix can be *factorized* into two matrices, one representing the rotation, and the other representing the shape of the object. A straightforward Singular Value Decomposition (SVD) of this matrix results in the recovery of this factorization only up-to a *generalized linear corrective transform* (§Eq. 3). Solving this linear transform is a non-trivial task for several reasons as has been recently observed in the literature.

Further, the current factorization based solutions are not directly adaptable to the human movement problem (our interest) since the quality of the solution degenerates very rapidly when the "deformations" are large[1] .

Contribution: In this paper we propose a novel *constrained factorization* algorithm, which effectively couples prior learned statistical knowledge about human shape variability (and the subspace it spans) from the ground truth motion capture data, with non-rigid factorization algorithm. Specifically, we make use of motion capture data to build a prior *reference pre-shape* (§Sec. 3.1) . We assume that the recovered shape from the NRF algorithm should be structurally similar to the reference pre-shape. This is formulated as a constrained non-linear programming problem. These constraints on the structure of shape subspaces reduces the search domain and renders the problem well-posed (Eq. 6). We provide qualitative and quantitative results to demonstrate the validity of our scheme.

Notation: We follow the notation used in [10]. a is a scalar, $\mathbf{a}$ is a vector and $\mathbf{A}$ is a matrix. $\otimes$ denotes Kronecker product. $\odot$ denotes Hadamard product. $\text{vec}(\mathbf{A})$ vectorizes $\mathbf{A}$ by stacking its columns and $\text{vech}(\mathbf{A})$ vectorizes only its lower triangular portion. $\mathbf{A}^{\dagger}$ denotes the generalized inverse. $\text{vc}(\mathbf{x},\mathbf{y}) = \text{vech}(\mathbf{xy^T} + \mathbf{yx^T} - \text{diag}(\mathbf{x} \odot \mathbf{y}))$. Note that $\text{vc}(\mathbf{x},\mathbf{y})$ operator helps to represent equations of the form $\text{vec}(\mathbf{x^T A y})$ when $\mathbf{A}$ is symmetric, more concisely as $\text{vc}(\mathbf{x},\mathbf{y})^{\text{T}}.\text{vech}(\mathbf{A})$

Road Map In Section 2 we outline two different applications of existing NRF methods, which are relevant in our context. We first describe how NRF can be used to de-noise and fill in missing entries of a noisy and possibly incomplete data sequence. This is followed up with a brief overview of a straightforward way of using prior NRF methods, with our experiments that exposes some problems. Section 3 formalizes our notion of *shape* and describes how shape variability of an ensemble of data can be captured. Section 4 gives the details of a Sequential Quadratic Programming based constrained optimization scheme which couples NRF algorithm with the learned statistical data. We discuss our experiments and results in Section 5 and conclude in Section 6.

2 Non Rigid Factorization

Apart from structure from motion, factorization techniques can be applied to a wide range of application like data segmentation, data de-noising and data imputation. Data de-noising and imputation are of significant interest to us since the feature tracks from the off-the shelf trackers are

[1] There has been some recent work on extending factorization methods for articulated structures [11, 12]. But these methods require a very large number of features, whereas we work with a very sparse number of features and assume the human body to be a deforming object

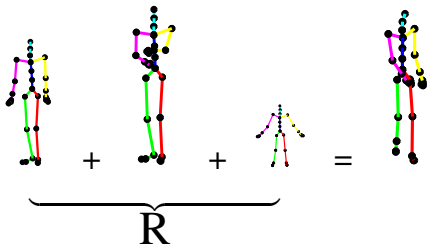


Figure 1: A pictorial representation of a *morphable model*. The right hand side is the actual data seen but can be obtained by modifying "basis" shapes.

typically noisy and contain missing information due to occlusion. The de-noising and structure recovering capability of the factorization algorithm is reviewed in this section.

The Basics: A popular representation for image formation (for either non-rigid or articulated objects) under orthographic or weak projective camera models is to write

$$\mathbf{W}_f = \mathbf{R}_f(\sum_{i=1}^{K} c_{fi}\mathbf{S}_i)$$

where $\mathbf{W}_f$ is the observed 2D feature in frame f (out of F given frames), $\mathbf{R}_f \in \mathbb{R}^{2\times 3}$ is the truncated row-orthonormal rotation matrix. K is the number of morph shapes needed to fully represent the object, $\mathbf{S}_i \in \mathbb{R}^{3\times P}$ the ith morph shapes (where P refers to the number of feature points tracked), and c_{fi}, the morph weights corresponding to $\mathbf{S}$ in the fth frame. This is pictorially represented in Fig. 1.

We build an *observation matrix* $\mathbf{W} \in \mathbb{R}^{2F\times P}$ by stacking the position of P landmark points observed in F frames. The structure of the observation matrix $\mathbf{W}$ appears in the left hand side of Eq. 1. Here (x_{ij}, y_{ij}) refers to the 2D co-ordinates of the point j in frame i.

$$\mathbf{P} = \begin{pmatrix} x_{11} & \cdots & x_{1P} \\ y_{11} & \cdots & y_{1P} \\ \vdots & \ldots & \vdots \\ x_{F1} & \cdots & x_{FP} \\ y_{F1} & \cdots & y_{FP} \end{pmatrix} = \mathbf{MS} = \underbrace{\begin{pmatrix} \mathbf{c}_1^T \otimes \mathbf{R}_1 \\ \vdots \\ \mathbf{c}_F^T \otimes \mathbf{R}_F \end{pmatrix}}_{2F\times 3K} \underbrace{\begin{pmatrix} \mathbf{S}_1 \\ \vdots \\ \mathbf{S}_F \end{pmatrix}}_{3K\times P} \tag{1}$$

This factorization can be performed modulo a *gauge factor* of $\mathbf{G} \in GL(3K, 3K)$ [8](§Sec.2.2) using SVD, if we assume an isotropic and Gaussian noise model[2]. But when there are outliers and missing data, which indeed is the case with most real-life measurements due to tracking failure and outliers, a straightforward SVD is no longer applicable.

2.1 Data denoising and missing information recovery

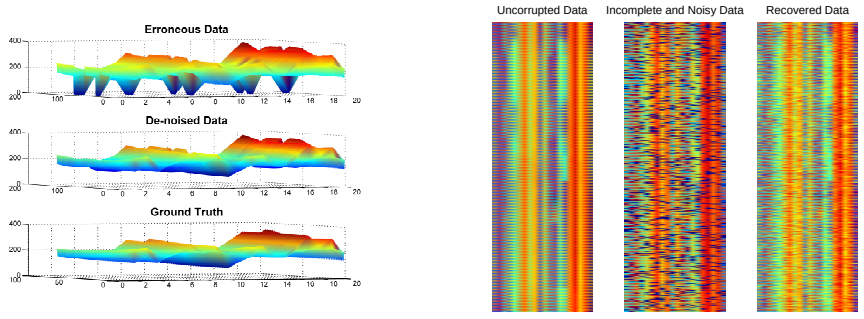
The most commonly used approach is to re-write the above problem with some robust ρ-function where the contribution of each item is weighted according to its fitness to the subspace [13, 14]. The modified factorization problem is now to compute the maximum likely estimator of a weighted L_2 norm cost function.

$$\varepsilon_{\text{mle}}(\tilde{\mathbf{M}}, \tilde{\mathbf{S}}) = ||\mathbf{W} \odot (\mathbf{P} - \tilde{\mathbf{M}}\tilde{\mathbf{S}})||_F^2 \tag{2}$$

where $w_{ij} \geq 0$ is a weighing factor which specifies the uncertainty in $\mathbf{p}_{ij}$ and $w_{ij} = 0$ if $\mathbf{p}_{ij}$ is missing

The literature on factorization with missing data falls into several categories: close-form solutions, imputation methods, EM-akin alteration methods and direct non-linear minimization methods. An excellent comparative study between these various method can be found in [14].

[2]Note that though the factorization assumes that temporal dependices in the data are caught by the tracker, the rank constraint enforces another layer of weak and subtle constraint on the contunity of motion.



(a) (b)

Figure 2: Surface and matrix plots (left and right hand side respectively) of noisy+incomplete data, de-noised data and Ground Truth. Notice that the recovered data has a high similarity to the ground truth

Our Denoising Method: We make use of the second order damped Newton algorithm introduced in [14] to de-noise the noisy point tracks. But we additionally perform modified Gram-Schmidt orthogonalization on the current estimate of both $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{S}}$ at each iteration. Note that Eq. 2 does not impose any structure on $\tilde{\mathbf{M}}$ or $\tilde{\mathbf{S}}$, whereas SVD based solutions ensured that $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{S}}$ are orthonormal and form good bases. We find that enforcing the orthonormality at each step makes the algorithm more numerically robust, rather than performing one single SVD toward the end. We initialize the optimization with left and right subspace estimate from a sparse SVD [15] of the incomplete data matrix. We weigh the visible features isotropically. These weights are estimated by contrasting the singular value spectrum of the sparse SVD with the mean value of a prior computed ensemble of spectrum of non-noisy, complete and typical data sets. The features which are not visible are assigned zero weights. A typical example is shown in Fig. 2(a). The deep trenches in the top figure corresponds to the missing data feature points. Observe that these valleys disappear after the de-noising step (middle figure). Moreover, the recovered data has a high similarity to the ground truth (bottom figure).

2.2 Recovering Motion and Shape

As mentioned earlier, unfortunately this factorization is not unique, but determinable only up to a non-singular linear corrective transformation $\mathbf{G}$ as

$$\mathbf{M} = \tilde{\mathbf{M}}.\mathbf{G} \qquad \mathbf{S} = \mathbf{G}^{-1}.\tilde{\mathbf{S}} \tag{3}$$

where we have the true scaled rotation matrix $\mathbf{M}$ and shape matrix $\mathbf{S}$. The heart of the non-linear factorization algorithm lies in solving for this corrective transform $\mathbf{G} \in \mathbb{GL}^{(3K\times 3K)}$ as described briefly below.

Let $\mathbf{x}_f^T$ and $\mathbf{y}_f^T$ be the pair of rows in $\mathbf{M}$ which gives the projection for frame f. Notice that $\mathbf{M}$ is made up of blocks of 2×3 scaled rotation matrices. Hence rows of each of these 2×3 blocks must be orthogonal and of equal norm.

$$\begin{aligned}
\mathbf{x}_f^T\mathbf{y}_f = 0 \quad &\text{(orthogonality constraint)}\\
\tilde{\mathbf{x}}_f^T\mathbf{G}\mathbf{G}^T\tilde{\mathbf{y}}_\mathbf{f} = 0 \Rightarrow \mathrm{vc}(\tilde{\mathbf{x}}_\mathrm{f},\tilde{\mathbf{y}}_\mathrm{f})\mathrm{vech}(\mathbf{G}\mathbf{G}^\mathbf{T}) = 0 \quad &\\
\mathbf{x}_f^T\mathbf{x}_f = \mathbf{y}_f^T\mathbf{y}_f \Rightarrow (\mathbf{x}_f-\mathbf{y}_f)^T(\mathbf{x}_f+\mathbf{y}_f) = 0 \quad &\text{(equal norm constraint)}\\
(\tilde{\mathbf{x}}_f-\tilde{\mathbf{y}}_f)^T\mathbf{G}\mathbf{G}^T(\tilde{\mathbf{x}}_f+\tilde{\mathbf{y}}_f) \Rightarrow \mathrm{vc}(\tilde{\mathbf{x}}_\mathrm{f}-\tilde{\mathbf{y}}_\mathrm{f},\tilde{\mathbf{x}}_\mathrm{f}+\tilde{\mathbf{y}}_\mathrm{f})\mathrm{vech}(\mathbf{G}\mathbf{G}^\mathbf{T}) = 0 \quad &\\
\textit{Let } \mathbf{L} = [\mathrm{vc}(\tilde{\mathbf{x}}_\mathrm{f},\tilde{\mathbf{y}}_\mathrm{f}),\mathrm{vc}(\tilde{\mathbf{x}}_\mathrm{f}-\tilde{\mathbf{y}}_\mathrm{f},\tilde{\mathbf{x}}_\mathrm{f}+\tilde{\mathbf{y}}_\mathrm{f})]^T\forall f \text{ and } \mathbf{Q_A} = \mathbf{L}\mathbf{L}^T \quad &
\end{aligned} \tag{4}$$

Note that $\mathbf{M}_{1:3} = \tilde{\mathbf{M}}\mathbf{G}_{\mathbf{1:3}} \in \mathbb{R}^{2\mathrm{F}\times 3}$. It turns out that solving for $\mathbf{G}_{1:3}$ is sufficient to solve for the rest of $\mathbf{G}$ [10]. The vanilla NRF computes $\mathbf{G}_{\mathbf{1:3}}\mathbf{G}_{\mathbf{1:3}}^{\mathbf{T}}$ that minimizes the sum squared deviation from orthogonality in the final motion matrix by least squares solving the system of equations given by

$$\mathrm{OrthErr}_{\mathbf{Q_A}}(\mathbf{G}_{1:3}) = \mathrm{vech}(\mathbf{G}_{1:3}\mathbf{G}_{1:3}^\mathrm{T})^\mathrm{T}\mathbf{Q}_\mathrm{A}\mathrm{vech}(\mathbf{G}_{1:3}\mathbf{G}_{1:3}^\mathrm{T}) \tag{5}$$

The symmetric matrix $\mathbf{G}_{1:3}\mathbf{G}_{1:3}$ is later decomposed to $\mathbf{G}_{1:3}$ by performing a rank-3 EVD ($\mathbf{G}_{1:3} = \mathbf{V}\Lambda^{0.5}$)

Significantly, it was recently shown [9] that these rotation constraints are not sufficient to *uniquely* solve for the corrective transform $\mathbf{G}$ for articulate and non-rigid motions. More specifically the general solution of the rotation constraints is $\mathbf{GHG}^T$, where $\mathbf{H}$ is the summation of an arbitrary block skew symmetric matrix and an arbitrary block scale identity matrix. The culprit being the redundancy in the constraint matrix which leaves the solution to Eq. 5 under-constrained. One way to overcome this ill-posedness is a heuristic scheme proposed by the authors of [9] where shapes in K frames are assumed to be independent and will act as a set of bases. Unfortunately, in general, this is not a good practice, since it tries to represent the shape space non-parsimoniously with a finite set of local diffeomorphisms, and hence has questionable subspace representation ability [16].

An alternative appears in [10] where Brand makes another relevant observations that approximation of Eq. 5 as a nested *linear* least square solution doesn't do justice to the physical reality. It overlooks a lot of co-variance information encoded in $\mathbf{Q}_\mathbf{A}$. Instead, the author solves $\mathbf{G}_{1:3}$ directly from Eq. 5 using a variant of first order line search global optimization framework (the step sizes are calculated by direct root finding). But, our experiments showed that the error surfaces generally have a rough terrain and many a times converge to the dreaded local minima. An example is show in Fig. 3.

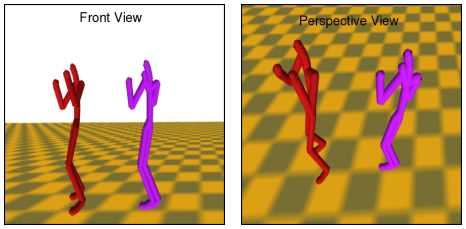


Figure 3: Non-rigid factorization algorithms have the tendency to flatten the body structure (notice the legs). The red colored human model is the representation of the actual data and the pink colored model is a reconstruction from 2D data.

The vanilla NRF, does not make any assumption about the shape of the object in scene. But a huge chunk of vision related engineering problems (in our case human pose extraction) do allow us to make *valid* assumption regarding object shape subspaces and possibly get an estimate of the subspace apriori. In the next section we describe how a good prior estimate of shape subspace can be obtained.

3 Shape Analysis

The word "shape" is very commonly used in everyday language, usually referring to the appearance of an object. Following Kendall [17] the definition of shape that we consider is:

> *Shape is all the geometrical information that remains when location, scale and rotational effects are filtered out from an object*

Important aspects of shape analysis are to obtain a measure of distance between shapes, to estimate average shapes from a random sample and to estimate shape variability from a random sample.

Procrustes analysis involves matching configurations with similarity transformations to be as close as possible according to Euclidean distance, using least squares techniques. More formally, given two mean centered configuration matrices $\mathbf{X}_1$ and $\mathbf{X}_2$, the *full Procrustes distance* between $\mathbf{X}_1$ and $\mathbf{X}_2$ is

$$\mathrm{D}_{\mathrm{pro}} = \inf_{\Gamma \in SO(3), \beta \in \mathbb{R}} ||\mathbf{Z}_2 - \beta \mathbf{Z}_1 \Gamma||$$

where $\mathbf{X}_r = \mathbf{Z}_r / ||\mathbf{X}_r||, r = 1,2$

Similarly, the full Procrustes estimate of mean shape $[\hat{\mu}]$ is obtained by minimizing (over μ) the sum of square full Procrustes distance from each $\mathbf{X}_i$ to an unknown unit mean configuration μ, i.e

$$[\hat{\mu}] = \arg\inf_{\mu} \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{d}_{\mathrm{F}}^2(\mathbf{X}_{\mathrm{i}}, \mu)$$

For a more detailed exposition, we refer the readers to [18] and the original work of Kendall [17]

3.1 Creating The Reference Pre-Shape

In the last decade or so, principal component analysis (PCA) has become a favorite tool for computer vision and graphics researchers [19, 20]. PCA is a simple, yet powerful technique to collect and investigate the statistically variability of data which resides in linear spaces ($\mathbb{R}^3$ in our case). To learn a good set of bases we need a corpus of accurate data with wide variability, which now a days is publicly available in the form of archival motion capture data.

Each pose is parametrized as a single observation 60 dimensional column vector ($\mathrm{vec}(\mathbf{Q}_{\mathrm{train}})$) containing the Euclidean positional information of all the land mark points[3]. We borrow techniques from Procrustes Analysis introduced in the previous section to strip these vectors of positional, scale, and orientation details.

If $\hat{\mu}$ be a pre-shape corresponding to the full Procrustes mean shape, the aligned vectors can be computed as

$$\mathrm{v}_{\mathrm{F}} = (\mathbf{1} - \mathrm{vec}(\hat{\mu}\hat{\mu}^{\mathrm{T}}))\mathrm{vec}(\hat{\beta}_{\mathrm{i}}\mathbf{Q}_{\mathrm{train}}\hat{\Gamma}_{\mathrm{i}})$$

These aligned vectors are stacked into a data matrix $\mathbf{X}_{\mathrm{mocap}}$ and we compute the principal components of this data matrix. PCA performs a basis transformation to an orthogonal co-ordinate system formed by the eigen vectors $\mathbf{V}_i$ of the covariance matrix of $\mathbf{X}_{\mathrm{mocap}}$. These orthogonal components are ordered with respect to the descending values of their eigenvalues and are arranged into $\mathbf{S}_{\mathrm{ref}}$. We call $\mathbf{S}_{\mathrm{ref}}$ as *Reference Pre-shape*. For a full body motion with just 5 bases we are able to represent more than 94% variation in the data.

4 Constrained Factorization

The primal idea behind our method is that shapes recovered by the NRF should having significant similarity to the pre-learned *Reference Pre-Shape*. We express this as a constrained non-linear programming problem.

More formally, we rewrite Eq. 5 as

$$\begin{array}{rcl} E(G_{1:3}) & = & \mathrm{vech}(\mathbf{G}_{1:3}\mathbf{G}_{1:3}^{\mathrm{T}})^{\mathrm{T}}\mathbf{Q}_{\mathrm{A}}\mathrm{vech}(\mathbf{G}_{1:3}\mathbf{G}_{1:3}^{\mathrm{T}}) \\ S.T & & \mathrm{trace}(\mathbf{G}_{1:3}\mathbf{G}_{1:3}^{\mathrm{T}}) = 1 \\ & & \mathrm{D}_{\mathrm{pro}}^2(\mathbf{G}_{1:3}, \tilde{\mathbf{S}}\mathbf{S}_{\mathrm{ref}}^{\dagger}) \leq -\mathrm{d} \end{array} \tag{6}$$

where $\mathrm{D}_{\mathrm{pro}}(\mathbf{X}, \mathbf{Y})$ gives the orthogonal Procrustes distance between $\mathbf{X}$ and $\mathbf{Y}$ and d is an user-set parameter, which specifies the tolerance level for the structural variation and defines the feasible area or the domain of the cost function (smaller the tolerance, narrower the feasible area) . In our experiments we used 0.2 as the threshold. Though it is tempting to decrease the tolerance, lesser tolerance makes the algorithm more prone to over-fitting (especially if the training set is not exhaustive enough).

Notice that both our cost function and constraints are non-linear. While the cost function is quartic, the constraints are of quadratic nature. Though constrained non-linear optimization (in general) is still an open problem, many efficient, but approximate numerical schemes exist [21] especially for relatively lower order cost function (quartic, in our case) and near linear constraint functions (quadratic). We make use of *Sequential Quadratic Programming*, a well known and used numerical solution for optimizing smooth non-linear cost functions under smooth non-linear constraints [21, 22]. It is Newton like in that it requires second derivatives of the cost function and potentially provides quadratic convergence.

The goal is to extremize a scalar cost function $E(\mathbf{x})$ subject to a vector of constraints $\mathbf{c}(\mathbf{x}) \leq 0$. (Note that inequality constraints can be treated at par with the equality constraint by assuming its

[3] Note that the ordering (or the meta-knowledge about it) of this vector has to be consistent with the 2D observation vector.

respective Lagrange multiplier vanishes whenever the inequality is not strict [21] and is strictly positive whenever the inequality is strict). Lagrange multipliers λ give an implicit solution.

$$\bigtriangledown E + \lambda \bigtriangledown \mathbf{c} = \mathbf{0} \text{ with } \mathbf{c}(\mathbf{x}) = \mathbf{0}$$

We resolve this iteratively starting from some initial guess b$\mathbf{x_0}$. We approximate the cost to second order and the constraints to first order at $\mathbf{x_0}$, giving a quadratic optimization sub-problem with linear constraints.

$$\min_{\delta \mathbf{x}} \left(\bigtriangledown E.\delta x + \frac{1}{2} \delta \mathbf{x}^T \bigtriangledown^2 f.\delta \mathbf{x} \right) |_{\mathbf{c}+\bigtriangledown \mathbf{c}.\delta \mathbf{x}}$$

This sub-problem has an exact linear solution

$$\begin{pmatrix} \bigtriangledown^2 E & \bigtriangledown \mathbf{c}^T \\ \bigtriangledown \mathbf{c} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \delta \mathbf{x} \\ \lambda \end{pmatrix} = - \begin{pmatrix} \bigtriangledown \mathbf{E} \\ \mathbf{c} \end{pmatrix} \tag{7}$$

We solve for $\delta \mathbf{x}$, update $\mathbf{x}_0$ to $\mathbf{x}_1 = \mathbf{x}_0 + \delta \mathbf{x}$, re-estimate derivatives and iterate to converge.

The first order and second order derivatives of the Lagrange function in Eq. 6 are given in Appendix A.

5 Experiments

Training Data: We use 700 frames from motion capture data included in the HumanEva dataset [5] for learning the pre-shapes. These frames are selected by randomly sampling from the training set provided in the dataset. Selected frames span poses from various set of human action like walking, boxing, making hand gestures etc.

Testing: We test the performance of our algorithm on synthetic data with ground truth included in the testing set of the HumanEva dataset, as well as videos which give us only 2D information.

Motion-Capture Based Synthetic Data: In any choice of motion clip (from the motion capture data base) we know the 3D positions. We synthetically created a two dimensional projection by randomly choosing a center of projection. To simulate tracking errors and the like, the resulting "features" are further corrupted by adding Gaussian noise and frames dropped randomly to simulate quantifiable error and occlusion errors in the tracking process. This constituted the process of creating the observation matrix. The incomplete and noisy observation matrix is denoised using the method described in Section 2. Recall that the output of factorization is only accurate up-to an arbitrary rotation and scale. So the error at each frame is defined to be the Procrustes distance from the recovered orientation to the ground truth. We compare the performance of our algorithm to that of the unconstrained case [10].

Fig. 4 shows ground truth (left) contrasted with the output of our algorithm (middle) and the unconstrained case (right). The recovered pose by the unconstrained algorithm is nearly planar (notice the stick figure's left arm piercing its torso). The newly introduced boundary conditions ensured that the recovered solution did not collapse into a degenerate solution unlike the unconstrained state, and is found to be quite similar to the ground truth.

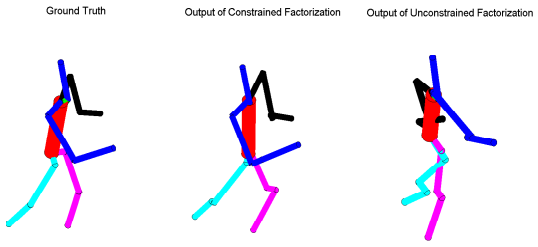
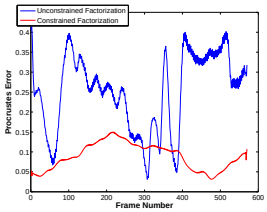


Figure 4: Ground truth data contrasted with the output from the constrained (our method) and non-constrained factorization (prior method) respectively.

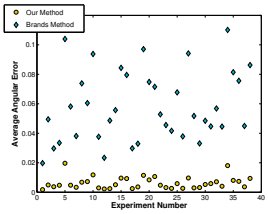
Next, we compared the performance of both algorithms over a novel long sequence (show in Fig. 5(a)). This sequence is novel in that it was not used for the computation of the reference

pre-shape. We selected a complicated clip of a boxing motion consisting of 577 frames sampled at 30Hz. The data is corrupted with 10% additive Gaussian noise and around 15% of its observations are masked out. Note that average performance of the constrained factorization algorithm hovers around the 5–15% reconstruction error mark. One interesting variation in the plot is that occasionally (frame numbers 290–320, 348–355 and 380–395) the error of the unconstrained algorithm dips somewhat below that of its constrained counter part (our method). The reason for this unexpected better performance is that during these frames, the actor is assuming a near planar pose and the degenerate shape base extracted by the unconstrained factorization algorithm is better able to explain these frames. Nevertheless, the unconstrained algorithm rapidly loses accuracy in the more common situation, when the actor resumes his or her flexible movements.

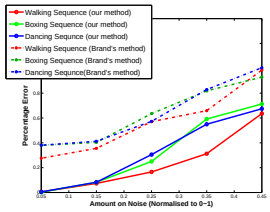
The scatter diagram in Fig. 5(b) plots the average error recorded by the constrained factorization algorithm (shown in yellow) and its unconstrained counterpart (shown in cyan) for various data input (a total of 39 different inputs). Each of the data input was seeded with 2% additive Gaussian error, and no occlusion condition was assumed. While carrying out these experiments we further assumed that the inequality constraints are strict. Fig. 5(c) shows the performance of both the version of the algorithm with three different sequence (walking , boxing and running) when subjected to different amount of synthetic noise. While the dotted line records the performance of the unconstrained version of the algorithm, the regular line record that of the constrained one. Walking, Boxing and Dancing motion sequence are represented by the red, green and blue lines respectively. Superior performance by the constrained version of the algorithm is amply recorded in every experiment.



(a) Error comparison of constrained and unconstrained NFM over a 577 frames boxing sequence



(b) Average error of constrained and unconstrained NFM from different experiments



(c) The performance of the algorithm with various amount of noise levels

Figure 5: Comparative Performance Evaluation

5.1 Data With No Ground Truth

In this experiment an 80 frame video sequence was semi-automatically tracked using the KLT based tracker. We hand picked the features which conformed to the anatomically relevant landmark points. We re-picked the lost features after every 10 frames. Note that far superior tracking schemes exist [23] for tracking humans from video. The purpose of this experiment was to test the performance under non-linear error models which often appear in real data sequences. Two different 'pigeon' views of the recovered orientation of the actor is shown along with actual data is show in Fig. 6. As a post-processing step, the recovered data is smoothed out using a Kalman smoother. More output including the video of the just explained experiment can be found at `http://www.cse.iitb.ac.in/appu/bmvc07/`

6 Conclusion and Future Work

We have given a novel constrained non-rigid factorization algorithm that extracts 3D human poses from 2D video sequences. Both qualitative and quantitative results were provided. Note that our method can be applied to any deforming data sequences (apart from human motion), provided accurate motion capture or similar high precision quantized data exists.

Future Work: The strength and weakness of factorization based techniques lies in its block based nature. This potentially rules out any online scheme. We are currently exploring the possibility of having a windowed scheme, thereby making the algorithm semi-online. We are also considering having an iterative refinement of reference pre-shape, hence equipping the algorithm

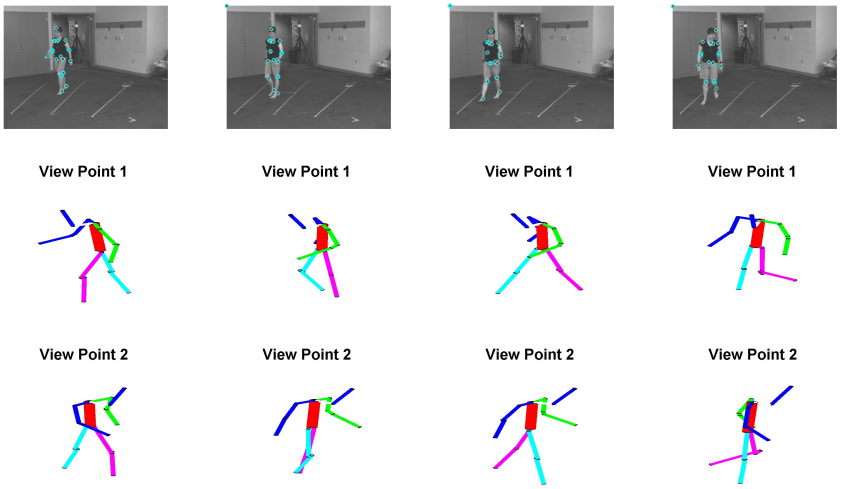


Figure 6: The top row shows the raw frames with features overlayed. The middle and bottom shows the recovered 3d pose rendered from two novel view points. The front view is identical and not shown.

to handle non-stationary data, and previously unseen data. Another possibility we wish to explore is to merge the optimization given in Eq. 2 and Eq. 6 as a single optimization problem.

A Derivatives

The corresponding Lagrange function of Eq. 6 can be written as

$$\begin{aligned}\mathscr{L} = \text{vech}(\mathbf{G}_{1:3}\mathbf{G}_{1:3}^{\text{T}})^{\text{T}}\mathbf{Q}_{\text{A}}\text{vech}(\mathbf{G}_{1:3}\mathbf{G}_{1:3}^{\text{T}}) + \lambda(\text{vec}(\mathbf{G}_{1:3})^{\text{T}}\text{vec}(\mathbf{G}_{1:3}) - 1) \\ + \mu(\text{vec}(\mathbf{G}_{1:3} - \tilde{\mathbf{S}}\mathbf{S}_{\text{ref}}^{\dagger}\Gamma)^{\text{T}}\text{vec}(\mathbf{G}_{1:3} - \tilde{\mathbf{S}}\mathbf{S}_{\text{ref}}^{\dagger}\Gamma) - \text{d}) \quad (8)\end{aligned}$$

where $\Gamma_i \in \mathbb{SO}(3)$. Let $\mathbf{Z} = \mathbf{G}_{1:3}$ and $\mathbf{J}_{ij} \in \{0,1\}^{3K\times 3}$ is all zeros except for element $J_{ij} = 1$

$$\begin{aligned}\frac{\partial\mathscr{L}(\mathbf{Z},\lambda,\mu)}{\partial Z_{ij}} =& 2\text{vech}(\mathbf{Z}\mathbf{Z}^{\text{T}})^{\text{T}}\mathbf{Q}_{\text{A}}\text{vech}(\mathbf{Z}\mathbf{J}_{\text{ij}}^{\text{T}} + \mathbf{J}_{\text{ij}}\mathbf{Z}^{\text{T}}) + \lambda\,\text{vec}(\mathbf{Z}\mathbf{J}_{\text{ij}}^{\text{T}} + \mathbf{J}_{\text{ij}}\mathbf{Z}^{\text{T}}) \\ &+ \mu(\text{vec}((\mathbf{Z} - \tilde{\mathbf{S}}\mathbf{S}_{\text{ref}}^{\dagger}\Gamma)\mathbf{J}_{\text{ij}}^{\text{T}} + \mathbf{J}_{\text{ij}}\text{vec}((\mathbf{Z} - \tilde{\mathbf{S}}\mathbf{S}_{\text{ref}}^{\dagger}\Gamma)^{\text{T}}) \\ \frac{\partial\mathscr{L}(\mathbf{Z},,\lambda,\mu)}{\partial\lambda} =& \text{vec}(\mathbf{G}_{1:3})^{\text{T}}\text{vec}(\mathbf{G}_{1:3}) \\ \frac{\partial\mathscr{L}(\mathbf{Z},,\lambda,\mu)}{\partial\mu} =& \text{vec}(\mathbf{G}_{1:3} - \tilde{\mathbf{S}}\mathbf{S}_{\text{ref}}^{\dagger}\Gamma)^{\text{T}}\text{vec}(\mathbf{G}_{1:3} - \tilde{\mathbf{S}}\mathbf{S}_{\text{ref}}^{\dagger}\Gamma) \\ \frac{\partial\mathscr{L}(\mathbf{Z},\lambda,\mu)}{\partial Z_{\text{ij}}Z_{\text{kl}}} =& 2.\text{vech}(\mathbf{Z}\mathbf{J}_{\text{kl}}^{\text{T}}) + \mathbf{J}_{\text{kl}}\mathbf{Z}^{\text{T}})\mathbf{Q}_{\text{A}}.\text{vech}(\mathbf{Z}\mathbf{J}_{\text{ij}}^{\text{T}} + \mathbf{J}_{\text{ij}}\mathbf{Z}^{\text{T}}) \\ &+ (\text{vech}(\mathbf{Z}\mathbf{Z}^{\text{T}})^{\text{T}}\mathbf{Q}_{\text{A}} + \lambda + \mu)\text{vech}(\mathbf{J}_{\text{kl}}\mathbf{J}_{\text{ij}}^{\text{T}} + \mathbf{J}_{\text{ij}}\mathbf{J}_{\text{kl}}^{\text{T}})\end{aligned} \quad (9)$$

References

[1] Soatto, S., Brockett, R.: Optimal structure from motion: Local ambiguites and global estimates. In: CVPR. (1998) 282–288

[2] Agarwal, A., Triggs, B.: Recovering 3d human pose from monocular images. IEEE Transactions on Pattern Analysis & Machine Intelligence **28**(1) (2006)

[3] Sigal, L., Black, M.J.: Predicting 3d people from 2d pictures. In: Articulated Motion and Deformable Objects, 4th International Conference. (2006) 185–195

[4] Taylor, C.J.: Reconstruction of articulated objects from point correspondences in a single uncalibrated image. Computer Vision and Image Understanding **80**(3) (2000) 349–363

[5] Sigal, L., Black, M.J.: Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. Technical Report CS-06-08, Dept. of Computer Science, Brown University, Providence, Rhode Island 02912 (2006)

[6] Ma, Y., Soatto, S., Košecká, J., Sastry, S.: An Invitation to 3-D Vision. From Images to Geometric Models. Springer (2004)

[7] Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization method. International Journal of Computer Vision (1992) 137–154

[8] Bregler, C., Hertzmann, A., Biermann, H.: Recovering non-rigid 3D shape from image streams. In: IEEE CVPR. (2000) 690–696

[9] Xiao, J., Chai, J.X., Kanade, T.: A closed form solution to non-rigid shape and motion recovery. In: ECCV. (2004) 573–587

[10] Brand, M.: A direct method of 3D factorization of nonrigid motion observed in 2D. In: Computer Vision and Pattern Recognition. (2005) 122–128

[11] Yan, J., Pollefeys, M.: A factorization-based approach to articulated motion recovery. In: CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2, Washington, DC, USA, IEEE Computer Society (2005) 815–821

[12] Tresadern, P., Reid, I.: Articulated structure from motion by factorization. In: Proc. 23nd IEEE Conf. on Computer Vision and Pattern Recognition, San Diego. (2005)

[13] Brandt, S.: Closed-form solutions for affine reconstruction under missing data. In: European Conference on Computer Vision, Springer-Verlag, 2002 (2002) 109–114

[14] Buchanan, A., Fitzgibbon, A.: Damped newton algorithms for matrix factorization with missing data. In: CVPR. Volume 2. (2005) 316–322

[15] Larsen, R.: Lanczos bidiagonalization with partial reorthogonalization. PhD thesis, Dept. Computer Science, University of Aarhus, DK-8000 Aarhus C, Denmark, (1998)

[16] Soatto, S., Yezzi, A.J.: DEFORMOTION: Deforming motion, shape average and the joint registration and segmentation of images. In: ECCV (3). (2002) 32–57

[17] Kendall, D.: Shape manifolds, procrustean metrics and complex projective spaces. Statistical Science **16** (1984) 81 – 121

[18] Dryden, I., Mardia, K.: Statistical Shape Analysis. Number ISBN 0-471-95816-6 in Wiley series in proability and Statistics. John Wiley and Sons (1998)

[19] Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active shape models and their training and application. Computer Vision and Image Understanding **61**(1) (1995) 38–59

[20] Blanz, V., Vetter, T.: A morphable model for the synthesis of 3d faces. In: SIGGRAPH, ACM Transaction on Graphics, New York, NY, USA, ACM Press/Addison-Wesley Publishing Co. (1999) 187–194

[21] Fletcher, R.: Practical Methods of Optimization. 2nd edition edn. John Wiley & Sons (1987)

[22] Triggs, B., McLauchlan, P., Hartley, R.I., A.W., F.: Bundle adjustment - a modern synthesis. In Triggs, B., Zisserman, A., Szeliski, R., eds.: Vision Algorithms: Theory and Practice, International Workshop on Vision Algorithms, Springer (1999) 298–373

[23] Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. International Journal of Computer Vision **61**(1) (2005) 55–79

Fast Motion Estimation on Range Image Sequences acquired with a 3-D Camera

Stephan Matzka[1,2] Yvan R. Petillot[1] Andrew M. Wallace[1]

[1] Heriot-Watt University, School of Engineering and Physical Sciences, Edinburgh, UK

[2] Ingolstadt University of Applied Sciences, Institute for Applied Research, Ingolstadt, Germany

Abstract

This paper presents a computationally efficient approach to estimate translational 3-D motion from range images sequences, that is adapted from a 2-D motion estimation algorithm. An implementation of the algorithm is evaluated for computational efficiency as well as robustness in the presence of noise for both synthetic and real-life range data acquired with a PMD device, a high-speed low-resolution 3-D camera.

1 Introduction

Motion estimation for intensity video images is well researched, with a number of proven concepts to create dense motion vector fields, possessing computational efficiency, or robustness against sensor noise. However, 3-D motion estimation on range images lacks fast and robust algorithmic concepts.

A current application field for translational 3-D motion estimation is given by the use of 3-D cameras in cars, such as a PMD camera [3]. In road traffic scenes, the only notable rotational motions are yaw movements which occur for cars bending off. Yet, even in this case, translational motion dominates due to the considerable turn radius of cars.

This paper presents a novel method to estimate translational 3-D motion from range images, that is adapted from a high-performance 2-D motion estimation algorithm. Its central qualities are computational efficiency and robustness in the presence of noise.

2 Related Work

The issue of estimating 3-D motion or optical flow fields from range images has been the subject of a number of publications. For example, an evaluation of 3-D motion estimation algorithms was given in Eggert et al., 1997 [2]. Many 3-D motion estimation approaches are based upon finding correspondences. These correspondences can be considered both local (cf. [1]), or global by solving a total least squares framework [8]. The resulting flow field of the latter method is dense, yet the complexity is high and real-time computation is not feasible with current hardware.

A correspondenceless approach was pursued by Liu and Rodrigues, 1999, based upon the cross matrix to estimate the motion parameters [6]. It is also possible to use the shift of previously segmented surfaces in a range image for motion estimation [5]. This approach is restricted to small relative motion between the camera and the scene and the segmentation process in itself is complex.

Apart from the cited work on 3-D motion estimation – which is only a selection – 2-D optical flow is a major topic of interest. Most of the 2-D motion estimation algorithms used in video-encoders are designed to be computationally efficient, which is also a constraint for real-time motion estimation.

However, to estimate 3-D motion in range images under real-time constraints, neither 2-D motion estimation based on difference measures nor 3-D motion estimation algorithms with high complexity can be used. Therefore, this paper suggests adapting a 2-D motion estimation algorithm for use on range images.

3 2-D Motion Estimation using PMVFAST

The Predictive Motion Vector Field Adaptive Search Technique (PMVFAST) is a block based motion estimation technique based upon MVFAST [4], which is an essential part of several video-coding standards, such as MPEG-1/2/4 [9]. PMVFAST has shown to be faster than other motion estimators while retaining a motion estimation quality comparable to a significantly slower full search algorithm.

PMVFAST uses a Diamond Search (DS) pattern (cf. Fig. 1a). Beginning in the centre, the (0,0) motion vector (MV) is the initial starting point. Then the search path is meandering circularly around the centre, performing a full orbit each time before increasing its search distance up until the maximum search distance.

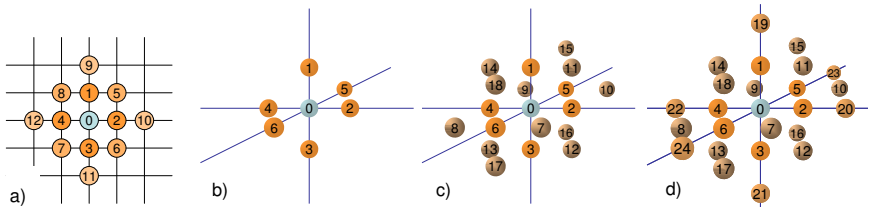


Figure 1: Fig. a) shows a Diamond Search pattern used for 2-D motion estimation. Exemplary PCS path building process: b) shows all (1,0,0) variations (#1-6), b) extends a) with all (1,1,0) variations (#7-18), and c) extends b) with all (2,0,0) variations (#19-24).

At each point on the search path, a block in the previous frame is matched against a block in the current frame. The block in the current frame is shifted by the (i,j) values of the search path. The quality of the match is determined by a distortion measure. A widely used distortion measure is the sum of absolute differences (SAD, see Eq. 1), which omits the multiplications necessary for mean squared error but has a similar performance [9]. Blocks used in this paper are 5×5 pixels, resulting in 25 summations per comparison. Also, MVs are not calculated for every pixel, instead a regular grid is used with the grid distance increasing logarithmically with the range image's size.

$$SAD_{DS}(v_x, v_y) = \sum_{i,j \in DS} \left| I_k(x+i, y+j) - I_{k-1}(x+v_x+i, y+v_y+j) \right| \qquad (1)$$

The search for the minimum SAD is performed with two differently sized diamonds in [9]. The expected magnitude of motion is estimated by examining three neighbouring

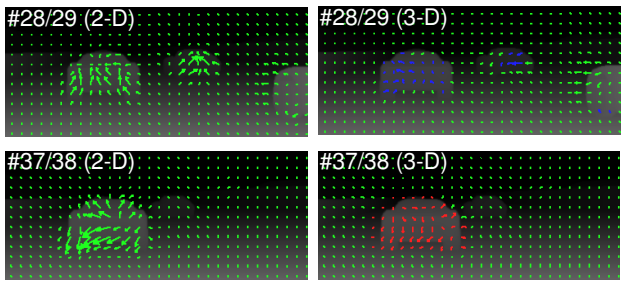


Figure 2: Motion vector fields of frame pairs #28/29 (increasing distance to car in front) and frame pair #37/38 (decreasing distance) in the Torcs sequence. Motion vector field (2D) shows the result using a 2-D full search algorithm, whereas (3D) shows the PCS result. Blue arrows indicate an increasing distance, red arrows a decreasing distance. The background shows the range images on which the motion estimation has been performed.

MVs at $(x-1,y)$, $(x,y-1)$, $(x+1,y-1)$, the previous MV at (x_{k-1},y_{k-1}), and the median MV. The average of these MVs is then used as an estimation for the current MV.

If the estimated MV for (x,y) is small, a small search diamond is used with the (0,0) MV as its centre. If the MV is estimated to be intermediate, a large diamond is used, again with (0,0) MV as starting point. In the case of high estimated motion, the small diamond is used with the estimated MV as its centre.

Regardless of the estimated motion, the (0,0) MV is examined first, and – if the distortion is below a chosen threshold – no further matching is done. Otherwise, the DS is performed and the displacement featuring the minimum distortion is chosen as the centre point in the next cycle. The search algorithm terminates if the centre of the search diamond is also the displacement with minimum distortion.

This concept holds for intensity images, yet in range images distance information is represented by intensity. On convex surfaces, such as a sphere, this will induce a difference-based 2-D motion estimation to detect a concentric outward motion if the distance is decreasing (it is implied, that small distances are represented by a high intensity), and a converging motion if the distance is increasing.

The above behaviour does not heavily affect MPEG motion estimation, since the aim there is not to calculate exact MVs, but to maximally reduce the video's bit rate while having as little visible quality loss as possible. However, for range images this effect leads to the necessity to consider depth motion in order to get accurate motion vectors.

4 Extending Diamond Search for use on Range Images

The idea of using a diamond shaped search path is extendible towards a 3-D translational motion estimation from range images. The least complex diamond shape in 3-D is a regular octahedron which will be referred to as *Point Cut Search* (PCS) path.

The PCS path will expand continuously, adding new layers around the origin in a point cut shape. The first layer has a distance of 1 to (0,0,0) and consists of the six permutations (1,0,0), (0,1,0), (0,0,1), (-1,0,0), (0,-1,0), and (0,0,-1) with varying signs.

The following base coordinates are (1,1,0); (1,1,1); (2,0,0); (2,1,0); (3,0,0) etc. These base coordinates are then permutated (maximum 6 permutations if all values are unique) with changing signs for every value (maximum 8 sign combinations if no value is zero). An illustration of the PCS path building process is given in Fig. 1b-d.

Both PMVFAST and PCS realise horizontal and vertical displacements by shifting the observation window in the actual frame horizontally and vertically. In PCS, displacements in distance in range images are represented as changes of intensity. Therefore, by adding or subtracting the value corresponding to the range displacement to the intensity values in the observation window, a displacement in distance can be modelled (see Eq. 2).

$$SAD_{PCS}(v_x, v_y, v_z) = \sum_{i,j,k \in PCS} \left| I_k(x+i, y+j) - I_{k-1}(x+v_x+i, y+v_y+j) + v_z + k \right| \quad (2)$$

As for PMVFAST, the search terminates when the centre point of the PCS is the point with minimum distortion or when the maximum number of iterations is reached.

5 Evaluation of the implemented Motion Estimator

The proposed motion estimator was implemented using four layers of abstraction (cf. Fig. 3). First, the range images are filtered in order to remove noise (temporal filtering using previous frames is optional). Second, subsequent filtered range images are searched for correspondences, using PCS. The resulting motion vectors are then filtered to remove outliers. Finally, the filtered motion vector field is used by PCS to predict the motion vectors for the next motion estimation.

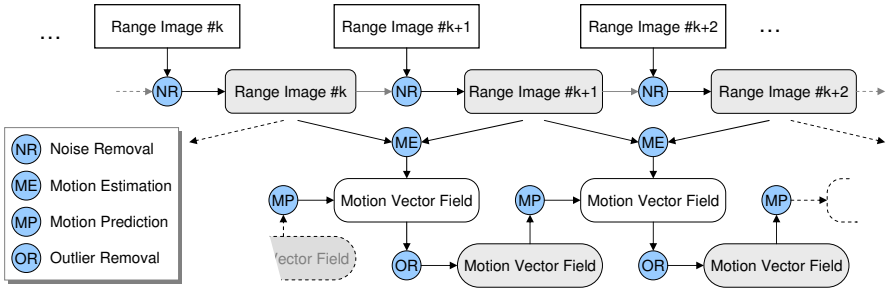


Figure 3: Block diagram of the implemented PCS motion estimator. Circles represent processing / filtering operations that are performed by the motion estimator, while boxes represent different abstraction layers from unfiltered range images to filtered MV fields.

5.1 Computational cost

The computational cost of the implemented motion estimator is evaluated using the simulated range image sequence extracted from *Torcs*[1]. The sequence consists of 155 frames

[1]Torcs is an open source racing game (`http://torcs.sourcforge.net`) using OpenGL. See supplementary video: `http://emfs1.eps.hw.ac.uk/~ceeyrp/BMVC2007/motionTorcs.avi` showing the range im-

recorded at 15 frames per second and a resolution of 500×220 pixel. Range is encoded with 8 bit, representing 256 range values, which is a coarse yet sufficient range resolution.

For each configuration, the average number of comparisons needed for each motion vector and the average SAD for the chosen motion vectors were taken as indicators of the computational cost and motion vector field quality respectively. To get a benchmark for these two values, a Full Search (FS) has been used (cf. Tab. 1).

Evaluating the performance of the PCS search strategy, the maximum number of iterations to shift the local minimum to the PCS's centre is the most important parameter. For evaluation, two PCS paths were chosen. The small PCS used has a maximum search distance of 2, the large PCS has a maximum search distance of 5. The threshold for (0,0,0) MV was set to 16, a value which yielded good results in the evaluation.

	FS	PCS_2	PCS_3	PCS_4	PCS_5	PCS_6	PCS_7
Comparisons per MV	75.52	19.72	22.49	24.70	25.72	26.48	27.10
Average SAD per MV	33.16	45.46	40.21	37.04	35.69	34.60	33.86
Efficiency Measure (Product)	2504.4	897.5	904.3	915.0	918.0	916.1	917.6

Table 1: Comparisons per MV and average SAD for motion estimation in the Torcs sequence. A full search (FS) is used as benchmark for the PCS_n with n maximum iterations. The efficiency measure is the product of comparisons per MV and average SAD.

Tab. 1 shows the performance of the PCS strategy with respect to the maximum number of iterations allowed. The lowest average SAD of 33.86 for 7 maximum iterations comes very close to the benchmark value $\overline{SAD}_{full}$ of 33.16, while needing only about a third (36%) of the comparisons.

In order to assess the efficiency of the PMVFAST search strategy, the product of comparisons needed for each MV and the average SAD is a possible metric. This product grows with increasing computational cost and distortion, for low computational cost and low distortion the product is small (cf. Tab. 1), the latter being true for PCS.

5.2 Quantitative Evaluation of Accuracy

A comparison of the estimated motion vector fields of a synthetic motion pattern against a ground truth known from the rendering process of the pattern has been conducted.

5.2.1 Motion Ground Truth

The motion pattern consists of two spheres diametrically orbiting around the range image's centre (x,y,z) = (160,120,127) so that the sphere in front occludes the sphere behind it intermittently. The underlying motion function for this pattern is

$$v_k = \begin{pmatrix} \lfloor 80.0 \cdot \sin(\frac{k}{30}) + 160.5 \rfloor \\ \lfloor 60.0 \cdot \cos(\frac{k}{30}) + 120.5 \rfloor \\ \lfloor 80.0 \cdot \cos(\frac{k}{30}) + 127.5 \rfloor \end{pmatrix} - \begin{pmatrix} x_{k-1} \\ y_{k-1} \\ z_{k-1} \end{pmatrix}, \; v_{max} = \begin{pmatrix} 3 \\ 2 \\ 3 \end{pmatrix} \tag{3}$$

The resulting range image sequence contains 200 frames with 320×240 pixels[2].

age, and the motion vector field estimated using PCS.

[2] See supplementary video: `http://emfs1.eps.hw.ac.uk/~ceeyrp/BMVC2007/motionOrbit.avi` showing the source range image, ground truth, motion estimation and motion vector field (from left to right).

5.2.2 Noise and Preprocessing Model

Range data sequences acquired by a 3-D camera suffer from a substantial amount of noise. This noise can be reduced by employing temporal filtering of a large number of frames. For traffic scenes, temporal filtering over a number of frames increases rotational motion of other traffic participants, which is not handled well by the algorithm.

In this trade-off between noise and rotational motion the algorithm has shown to be more capable of handling noise in the range images, therefore a diminutive number of frames for temporal filtering has been chosen.

The noise that occurs in 3-D camera range data sequences is best characterised as clipped Gaussian noise, as no negative distances or distances above the maximum measurable distance can appear, yet the distribution of noise suggests a Gaussian distribution (cf. Fig. 4). Therefore, the synthetic range image sequence has added noise of Gaussian distribution, where $0.0 \geq z(x,y) + z_{noise} \geq 255.0$.

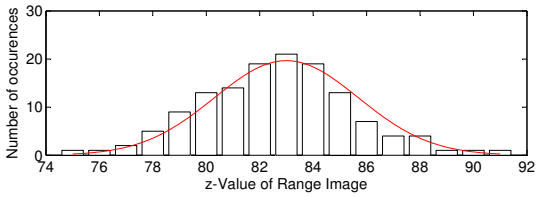


Figure 4: Distribution of range measurements of a constant distance over 135 frames (bars), which can be approximated by a Gaussian distribution with σ = 2.7 (red line).

Assuming a Gaussian noise model, spatial filtering using a Gaussian filter with $0.8 \geq \sigma_{RI} \geq 4.8$ presents a suitable preprocessing (cf. Fig. 5).

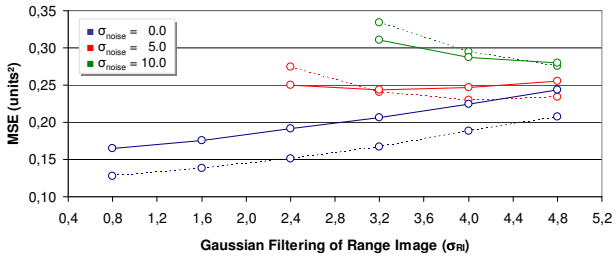


Figure 5: MSE of motion vector components for the orbiting movement pattern under influence of Gaussian noise σ_{noise} estimated by PCS_3 (solid line) and FS (dotted line) as compared to ground truth. The range image is processed using a Gaussian filter with σ_{RI}.

In Fig. 5, three major effects can be observed. First, if a noise-free range image is processed with a Gaussian filter, the MSE deteriorates as could be expected. Second, if a noisy range image is processed with a Gaussian filter, the MSE improves until a point where the range image is quasi noise-free and then shows the same behaviour as a noise-free image, that is MSE deterioration for higher standard deviations.

The third observable effect is, that PCS has a lower MSE for range images with a high

remaining noise after preprocessing. The reason for that is differing termination conditions. If a high level of noise is present during motion estimation, the correct MV does not necessarily exhibit the lowest SAD value. Using a full search, every displacement has the same probability to be selected as the estimated MV, whereas the iterative shifting in PCS makes it more probable, that a displacement near the initial starting point is selected.

The synthetic scene contains a large fraction of (0,0,0) MVs, therefore an incorrect MV close to an initial (0,0,0) MV starting point does not affect the MSE as much as a large MV, that is more probable to occur using a full search. However, it can be seen in Fig. 5 that this effect disappears when a suitable level of filtering is applied, so that the correct MSE exhibits the minimum SAD.

5.2.3 Regularisation Model

An analysis of the resulting MV fields against the ground truth suggest, that the main reason for high MSE values of the estimated motion vector fields is outliers caused by noise in the range image, not generic false motion vector estimation. Suitable methods to achieve outlier reduction include Gaussian or median filtering of the MV field.

In Fig. 6, MSE values for the same synthetic range image sequence as in Fig. 5 when using a Gaussian ($\times$) or median (Δ, using a 5×5 field) filter are shown.

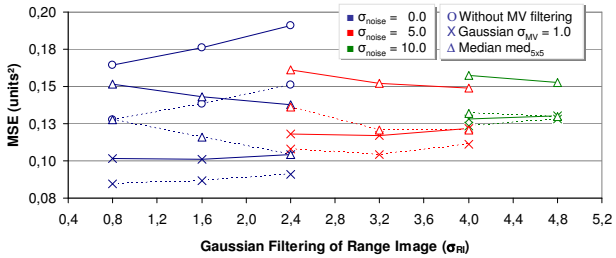


Figure 6: MSE of motion vector components for the orbiting movement pattern under influence of Gaussian noise σ_{noise} estimated by PCS_3 (solid line) and FS (dotted line) as compared to ground truth. The source range image is filtered using a Gaussian filter with σ_{RI}. The motion vector field is filtered using a Gaussian ($\times$) or median (Δ) filter.

In can be seen from Fig. 6, that the optimum MSE values gained by PCS at different levels of noise in the range images (including no noise) are within a narrow field (that is 0.1015 to 0.1616). Thich is an indicator that the algorithm is robust towards noise, if both input range images and motion vector fields are suitably filtered. The results are also comparable with the results gained by FS. At the same time, PCS computed the 320 $\times$ 240 pixels range image sequence at 11.8 frames per second (fps) on a standard 2.0 GHz PC, where FS performed at 1.85 fps, thus being more than six times (6.38) slower.

5.3 Performance on Data acquired with a 3-D Camera

In addition to synthetic range image sequences, the proposed algorithm has been evaluated using real-life data acquired by a PMD device, a high-speed low-resolution 3-D camera.

The 3-D camera is mounted inside the car close to the rear-mirror, observing an angle range of $55^\circ \times 18^\circ$ in front of the car. It acquires 64×16 pixel range images for distances up to 30m with a frame-rate of $\geq$100Hz [3]. In order to acquire a ground truth, a 2-D laser-scanner mounted on the car's radiator grille was used (see Fig. 7).

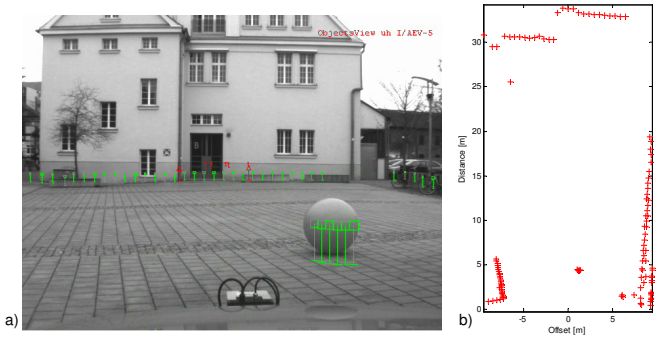


Figure 7: The left image a) shows the scene at frame #310 as seen from a grayscale intensity camera mounted close to the PMD device. The scatterplot b) on the right side shows the readings of the 2-D laser-scanner at the same frame.

As the proposed algorithm is designed to estimate translational motion, a large rubber ball is used due to its rotational invariance. Moreover, it is possible to reconstruct the ball's 3-D shape from the measured 2-D scan-line at any time, as the ball's radius and the scan-line's height are both known. In the scene, the ball is pushed in front of the stationary car and – due to a slightly inclined ground plane – performing a curve to the left, heading back towards the car (cf. Fig. 8a).

In order to determine the trajectory of the ball's centre, the readings of the laser-scanner are discarded unless they fall into a rectangle (distance 0..15m, offset -5..5m), which exclusively returns readings showing the ball. These readings fall onto a circle with the ball's radius. Thereby the ball's centre is determined fulfilling the circle equation Eq. 4 for the selected laser readings $(x_{reading}, y_{reading})$.

$$x_{centre}, y_{centre} = \arg \ (x_{reading_{1,2,..,n}} - x_{centre})^2 + (y_{reading_{1,2,..,n}} - y_{centre})^2 \qquad (4)$$

It is obvious, that Eq. 4 is overdetermined for $n > 2$, which can be solved by averaging all centre positions which are calculated using 2 laser readings at a time. The centre positions are then processed by applying both median and Gaussian filters in order to get a continuous motion (see Fig. 8a).

The range image sequence of the same scene is acquired with a PMD device[3] (see Fig. 8b). In order to be used with PCS, the range data has to be filtered over a small number of frames and outliers have to be rejected. Spatial filtering is not performed at this point, as the motion estimation algorithm includes this operation.

Generating a motion ground truth from the laser readings requires a calibration function from (x_{laser}, y_{laser}) to $(x_{pmd}, y_{pmd}, z_{pmd})$, which is approximated using a L_2 regression. (cf. [7]).

[3] See supplementary video `http://emfs1.eps.hw.ac.uk/~ceeyrp/BMVC2007/motionPMD.avi` showing the source range image, ground truth, motion estimation and motion vector field (from top to bottom).

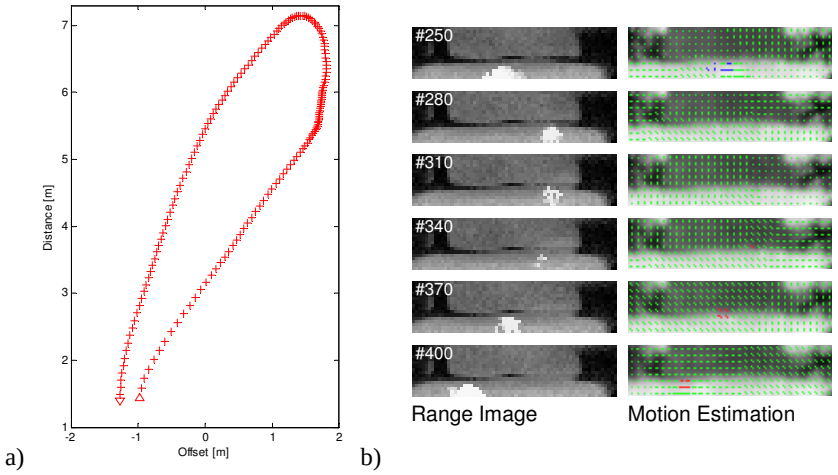


Figure 8: Scatterplot a) shows the ball's trajectory as detected with a laser scanner (Δ represents frame #250, ∇ frame #400). The range image sequence b) shows selected frames of the scene as seen by the PMD device (ball is brightened manually as to enhance visibility in the range image) as well as the corresponding estimated motion vector field. In the latter, blue arrows indicate an increasing distance, red arrows a decreasing distance.

The motion ground truth can now be generated from the ball's centre position. In Fig. 9 the MSE values of the motion estimation for the acquired range image sequence as compared to ground truth are shown.

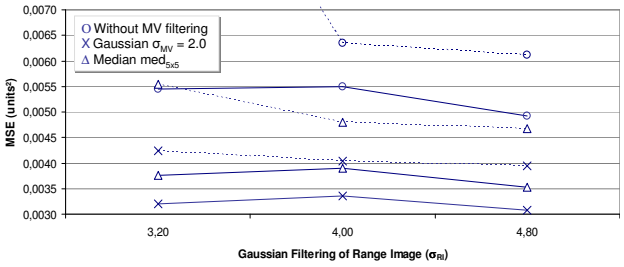


Figure 9: MSE of motion vectors components estimated by PCS_3 (solid line) and FS (dotted line) as compared to the ground truth under influence of Gaussian noise σ_{noise} for the orbiting movement pattern. The source range image is processed using a Gaussian filter with σ_{RI}.

Fig. 9 shows, that Gaussian or median filtering of the motion vector field results in a considerable reduction of the MSE. Both PCS and FS show small MSE values. Due to the large fraction of (0,0,0) MVs in the ground truth, the FS suffers from normal distribution of incorrect MVs in the presence of unfiltered noise, which is discussed in section 5.2.2 above. Again, PCS (46.9 fps) performed significantly faster than FS (19.5 fps) at a comparable motion vector quality.

6 Conclusion and Future Work

This paper presented a novel method to efficiently determine 3-D translational motion vectors in a range image sequence. The motion estimation has been evaluated on noisy, synthetic, and real-live range image data acquired by a PMD device and shown to be robust if a suitable filtering is applied on both range image and motion vector field.

Yet, there remain limitations for the proposed algorithm, which are largely those of PMVFAST. First, occlusion boundaries with little contrast between foreground object and background can lead to a motion vector pointing from the previous scene's background towards the occluding object's surface and vice versa. Second, rotational movements of objects must not be fast in order to find correct correspondences, which is generally true when using a high-speed 3-D camera on a road traffic scene. However, there still exists a trade-off between rotational motion and noise in range image sequences.

It has been shown that the computational cost for the acquisition of the motion vectors is low when compared to a full search. At a comparable motion vector field quality, PCS is shown to require only 16% – 42% of the number of comparisons a full search performs.

Future work will include evaluating the algorithm allowing a dynamic road-traffic range image scene as opposed to a static background and a fixed camera position. We should also evaluate other alternatives to the full search algorithm such as range flow, phase correlation or the use of a correlation-based matching criterion instead of a difference-based SAD measure.

References

[1] Krishnendu Chaudhury, Rajiv Mehrotra, and Cid Srinivasan. Detecting 3-d motion field from range image sequences. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 29(2):308–314, 1999.

[2] D. Eggert, A. Lorusso, and R.B. Fisher. Estimating 3-d rigid body transformations: a comparison of four major algorithms. *Machine Vision and Applic.*, 9:272–290, 1997.

[3] B. Fardi, J. Dousa, G. Wanielik, B. Elias, and A. Barke. Obstacle detection and pedestrian recognition using a 3d PMD camera. In *IEEE Intelligent Vehicles Symposium*, 2006.

[4] P.I. Hosur and K.K. Ma. Motion vector field adaptive fast motion estimation. In *Second International Conference on Information, Communications and Signal Processing*, 1999.

[5] X. Jiang, S. Hofer, T. Stahs, I. Ahrns, and H. Bunke. Extraction and tracking of surfaces in range image sequences. In *Proceedings of the 2nd International Conference on 3-D Digital Imaging and Modeling*, pages 252–260, 1999.

[6] Yonghuai Liu and Marcos A. Rodrigues. Correspondenceless motion estimation from range images. In *Proceedings of the Seventh International Conference on Computer Vision (ICCV'99)*, volume 1, pages 654–660. IEEE Computer Society, 1999.

[7] P. J. Rousseeuw and A. M. Leroy. *Robust regression and outlier detection*. John Wiley & Sons, Inc., New York, NY, USA, 1987.

[8] H. Spies, B. Jähne, and J. L. Barron. Range flow estimation. *Computer Vision Image Understanding*, 85(3):209–231, 2002.

[9] Alexis Michael Tourapis, Oscar C. Au, and Ming Lei Liou. Predictive motion vector field adaptive search technique (PMVFAST) - enhancing block based motion estimation. In *Proceedings of Visual Communications and Image Processing*, 2001.

A Practical Approach for Super-Resolution using Photometric Stereo and Graph Cuts

Swati Sharma and Manjunath V. Joshi
Dhirubhai Ambani Institute of Information and Communication Technology
Gandhinagar, India
{sharma_swati, mv_joshi}@daiict.ac.in

Abstract

In this paper, we propose an approach to obtain super-resolved image and super-resolved depth map using photometric cue. The images are captured using different light source positions which are assumed to be known. The surface of the object is assumed to be Lambertian. We model the high resolution structure (surface gradients) as a Markov Random Field (MRF) and use graph cuts with discontinuity preservation to get a high resolution depth map. We then reconstruct the high resolution intensity map for each light source position using the high resolution surface gradients. Results of experimentation on synthetic and real data are presented. The advantage of the proposed approach is that its time complexity is much less as compared to the super-resolution approaches that use global optimization techniques such as simulated annealing. Also, since we are using photometric cue, there is no need of registration as is required in motion based approaches.

1 Introduction

Many existing vision applications require high spatial resolution images to take better decisions. Since the resolution of an image is dependent on the device which is used to acquire the image, it is difficult to use very high resolution sensors as they are often expensive. Hence, there is a need to develop efficient methods to obtain better quality high resolution images given the low resolution observations. Also, 3-D shape recovery of a scene is used extensively for applications such as object tracking and recognition. Super-resolution is the process of obtaining a high resolution image from several low resolution images of the same scene. Most researchers use motion cue to increase the resolution of an image. Although the 3-D structure of the scene being imaged is inherently available from the disparity map, the motion cue being a 2-D feature matching technique does not consider the 3-D structure. Hence, techniques to obtain high resolution images which preserve the structure are required [2]. This motivates us to look into the use of photometric cue in order to estimate the shape of the object and the high resolution image.

For practical vision applications, high resolution depth and intensity map estimation methods which are computationally efficient are required. However, since the problem is ill-posed, many researchers use regularization based approaches in order to obtain better estimates. Now, if the cost used for obtaining the solution is non-convex (discontinuity preserving cost), then optimization techniques such as simulated annealing are used to

obtain the global minima, which makes these methods very time consuming. For instance, if we consider an assembly line where an object has to be moved from one place to another (industrial inspection), the requirement is to be able to calculate depth fast enough so that the assembly line functions smoothly. Here the requirement is the speed and not very high accuracy. In such situations near global optimization methods such as graph cuts are useful.

In [8] the authors show the estimation of super-resolved image and depth map using photometric cue. They model the surface gradients and albedo as the Markov Random Field's (MRF) and use line fields for discontinuity preservation. They also use additional constraints for optimization. Since they use simulated annealing for minimization, the approach is computationally very taxing.

In this paper, we solve the problem of simultaneous estimation of the super-resolved depth map and intensity map using photometric cue. We use graph cuts for optimization which is much faster as compared to simulated annealing minimization approach even when a discontinuity preserving cost function is used. Our results show that the performance (both perceptually and quantitatively) of graph cuts based approach for super-resolution is better than general interpolation techniques. The results also show that our super-resolution approach takes much less time as compared to the approach using simulated annealing.

2 Previous Work

The idea of super-resolution was first proposed by Tsai and Huang [12]. In literature, many researchers have proposed approaches for super-resolution that use motion as a cue. In [4], Ur and Gross use the Papoulis-Brown generalized sampling theorem to obtain a high resolution image from several low resolution spatially shifted images. A set theoretic approach to the super-resolution restoration that is based on iterative back projection method adapted from computer-aided tomography was proposed in [6]. Here, the output image is initialized and the temporary results are projected to the measurements (by simulation). The temporary results are updated according to the simulation error. A regularized constrained total least squares based approach to obtain high resolution image was proposed in [7]. Cheeseman et al. [9] use a Bayesian method for reconstructing a super-resolved surface model by combining the information from a set of given images. They find the "emmitance" of the surface which is a combination of albedo, illumination conditions and ground slope for landsat images. In [14], the authors consider graph cuts optimization for super-resolution using motion cue. The high resolution image is modeled as MRF and graph cuts is used for optimization to get the super-resolved image.

Researchers have also explored the possibility of super-resolving the intensity map of the scene as well as the depth map. The authors in [5] formulate the problem of super-resolution depth reconstruction as that of expectation maximization and use a probabilistic approach using MRF modeling. In [3], Shekarforoush et al. use MRFs to model the images to obtain high resolution depth and albedo from a sequence of displaced low resolution observations. The effect of sampling a scene at a higher rate is acquired by having sub-pixel displacements.

In [13], graph cuts minimization technique has been used for estimation of the surface normals using photometric stereo. They use the ratio of two images, in order to cancel

out the albedo in the image irradiance equation, and get the initial estimates of the surface normal which are required to define the energy functions. Graph cuts is then used for optimization. Tan et al. proposed a technique in [10] for enhancing the resolution for photometric stereo. Their method first uses the generalized reflectance model to recover the distribution of surface normals inside each pixel, from which they infer sub-pixel surface geometry on a surface by spatially arranging the normals among pixels at a higher resolution.

3 High Resolution using Photometric Stereo

If a Lambertian surface is assumed, the image irradiance equation relating the surface gradients and image intensity can be written as,

$$E_l(x,y) = R(p_l(x,y), q_l(x,y)) = \rho_l(x,y)\hat{n_l}(x,y).\hat{s} \tag{1}$$

where $p_l(x,y), q_l(x,y)$ are the surface gradients in (x,y) directions respectively. Here $\rho_l(x,y)$ represents the albedo, which is nothing but the fraction of light reflected from the surface at the point (x,y) and its value lies between 0 and 1. $\hat{n_l}(x,y)$ denotes the surface normal given by $\frac{(-p_l(x,y),-q_l(x,y),1)}{\sqrt{p_l(x,y)^2+q_l(x,y)^2+1}}$ and $R(p_l(x,y), q_l(x,y))$ is the reflectance map, $E_l(x,y)$ is the image irradiance (or intensity) at point (x,y) in the image. $\hat{s} = \frac{(-p_s,-q_s,1)}{\sqrt{p_s^2+q_s^2+1}}$ is a unit vector in the direction of the light source. Here, the subscript l denotes low resolution.

It has been shown in [2] that generalized interpolation can be used with photometric stereo to obtain high resolution. The high resolution image can be reconstructed using the interpolated values of the surface gradients and albedo using Eq.(1). This technique is called generalized interpolation. The advantage of using photometric cue for obtaining high resolution observations is that since there is no relative motion between the scene and the camera, the need for image registration with sub-pixel accuracy is eliminated.

In this paper we use graph cuts optimization which considers the spatial dependency with discontinuity preservation. Our algorithm converges much faster than simulated annealing and hence can be applied in a practical scenario. It may be noted here that we do not optimize for albedo assuming that it is a smooth field and a simple interpolation method can be used to interpolate albedo, while combining high resolution surface gradients and albedo to get high resolution intensity image.

4 Proposed Approach for Super-resolution using Graph Cuts

Typically, in any reconstruction based super-resolution technique, the available information from a number of low resolution observations is used together to get a single super-resolved image. First, a forward model is defined to establish the low resolution image formation process which is then used to establish a relation between the desired high resolution image and the given low resolution images. On the basis of this relationship the high resolution image is then obtained using an inversion process. The inversion process being ill-posed, requires the use of regularization and a suitable optimization approach such as the one proposed here can be used to minimize the derived cost function to obtain

better estimates. It is shown in [8] that it is indeed possible to obtain super-resolution using low resolution observations captured at different source positions.

4.1 Image Formation Model

Let E_{l_m} be the vector containing the intensity values of the mth low resolution image of size $M \times N$ arranged in lexicographical order and of size $MN \times 1$, where, $m = 1 \ldots K$. K is the number of available images. Similarly, let $\hat{n}$ and ρ be the vectors that represent the *high resolution* surface normal and *high resolution* albedo arranged lexicographically. Now, if D is the decimation matrix which represents the aliasing due to under sampling, the low resolution image formation model can be expressed as,

$$E_{l_m} = DH\rho(\hat{n}.\hat{s}_m) + w_m \tag{2}$$

Here, $\hat{s}_m$ represents the light source position for the mth image. w_m is the independent and identically distributed (i.i.d) Gaussian distributed noise vector with variance σ_w^2. H represents the blurring matrix. In our implementation, we assume H as an identity matrix and we do not consider blurred observations. We choose the decimation matrix as the average of the corresponding pixels of the high resolution image as given in [11].

4.2 Cost Function Formation

Regularization is a popular method for solving computer vision problems which are ill-posed. The approach consists of minimizing a cost function which is a sum of two terms i.e. data fitting term and regularization term [1]. In order to form the regularization term, we model the surface gradients (p and q) as MRFs. With the image formation model expressed in Eq. (2), it is quite simple to write the cost function to be minimized for estimating the high resolution entities as,

$$\varepsilon = \sum_{m=1}^{K} ||E_{l_m} - D\rho(\hat{n}.\hat{s}_m)||^2 + \sum_{a=0}^{MN-1} [\lambda_p min(|p_a - p_b|, T_p) + \lambda_q min(|q_a - q_b|, T_q)] \tag{3}$$

The first term in the cost function is called the data cost that measures the deviation from the observed data, caused by assigning a particular label (here surface gradients in the x and y directions) to a pixel. The other two terms are discontinuity preserving MRF priors for two neighboring pixels a and b. Here, p_a and q_a represent the labels assigned to a pixel a. p and q are labels of the surface gradients in the x and y directions respectively. T_p and T_q are thresholds that are used for discontinuity preservation.

The data cost [first term of Eq. (3)] at pixel a_1 of the high resolution image is given as follows,

$$Data(a_1) = \sum_{m=1}^{K} (E_{l_m}([\frac{a_1}{r}]) - \frac{1}{r^2}(F(a_1) + \ldots + F(a_{r^2}))^2 \tag{4}$$

where $[.]$ represents the integer value. The function $F(a_1)$ represents the term $\rho(a_1)(\hat{n}(a_1).\hat{s}_m)$ for a particular pixel a_1. $F(a_1), F(a_2) \ldots, F(a_{r^2})$ are the pixel intensities of the r^2 pixels, $(a_1, a_2, \ldots, a_{r^2})$ of the high resolution image related to one pixel of the low resolution image according to the matrix D. For instance, if the up-sampling factor is 2, the pixel $(0,0)$ of the low resolution is related to the pixel locations $(0,0)$, $(0,1)$, $(1,0)$ and $(1,1)$ of the

high resolution image. Hence, the data cost for the pixel $(0,0)$ (and also for pixels $(0,1)$, $(1,0)$ and $(1,1)$) of the high resolution entity to be estimated is,

$$Data(0,0) = \sum_{m=1}^{K} (E_{l_m}(0,0) - \frac{1}{4}(F(0,0)+F(0,1)+F(1,0)+F(1,1)))^2 \quad (5)$$

In order to use the graph cuts formulation for optimization the cost function should be regular. Applications of graph cuts generally use the data term that is a function of a single pixel [15]. Thus, in order to apply the graph cuts formulation we use valid mathematical approximations. It can be observed from the cost function that image intensities of several pixels of the high resolution image are related to the image intensity of a single pixel in the low resolution image in the data cost. So, we treat the remaining r^2-1 terms $(F(a_2),F(a_3),\ldots,F(a_{r^2}))$ as constant for a particular optimization step. Then the modified data term can be written as follows,

$$Data(a_1) = \sum_{m=1}^{K} (E_{l_m}([\frac{a_1}{r}]) - \frac{1}{r^2}(F(a_1)+C))^2 \quad (6)$$

where $C = F(a_2)+\ldots+F(a_{r^2})$.

So, the modified total cost function at a pixel a can be expressed as,

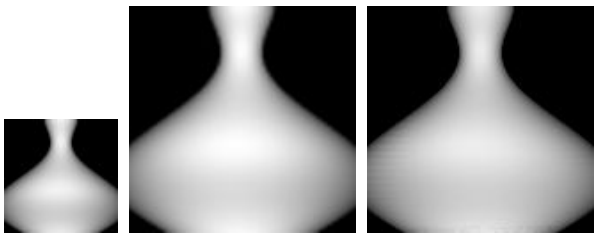
$$\varepsilon = \sum_{a=0}^{MN-1} [\sum_{m=1}^{K} (E_{l_m}([\frac{a}{r}]) - \frac{1}{r^2}(F(a)+C))^2 + \lambda_p.min(|p_a-p_b|,T_p)) + \lambda_q.min(|q_a-q_b|,T_q)] \quad (7)$$

where b is a neighboring pixel of a. The constant C represents the sum of the remaining r^2-1 terms, which are treated as constant for a particular optimization step.

We now optimize for the surface gradients, p and q using the graph cuts optimization. While optimizing for p field we consider q field as constant and vice versa. Both these fields are optimized one after another in each cycle until convergence is reached. The initial values for the high resolution p, q and ρ are obtained by interpolating the low resolution p, q and ρ fields (obtained using photometric stereo) by using a simple interpolation technique.

5 Choice of the Label Set

Graph cuts optimization requires a discrete label set. Most of the proposed methods that use graph cuts for optimization use integer labels. In our case, we use discrete floating point labels. Given the initial values of the surface gradients p and q, the range in which these fields lie is roughly known. Now based on the frequency distribution (histogram) of these labels it is possible to non-uniformly quantize the entire range of continuous values to get a discrete label set. The non-uniform quantization is done to assign maximum number of labels (discrete and integer) to that sub-range which has a higher probability. The number of labels, in this case, is directly related to the precision. As the chosen number of labels is increased, more accurate results may be obtained with a slight increase in computational complexity.



(a) (b) (c)

Figure 1: (a) Synthetically generated low resolution Vase image with light source position $(0,0,1)$ (b) Up-sampled image reconstructed using bi-cubic interpolation of p, q and ρ fields (c) Super-resolved Vase image using proposed approach

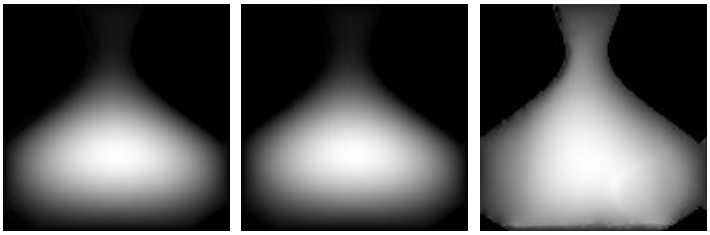
6 Experimental Results

In this section we present some of the results of our experiments. In order to test the performance of our algorithm, we first show results on a synthetic image Vase and then on a real image of a soft toy Jodu. We use the graph cuts library provided by Kolmogorov [18], [16], [17] with expansion algorithm for implementation.

First we consider the synthetic image Vase. Ten images of Vase of size 64×64 were generated using a computer program where each image is produced using a different light source position. These images are the given low resolution images. Now, in order to use graph cuts for optimization we need to use a fixed set of labels for each of the entities p and q. We observed that the initial values of p for the Vase image lie in the range $(-0.4, 0.6)$ and that of q lie in the range $(-0.2, 0.4)$. Hence, depending on the frequency distributions of the respective entities, we use 338 labels for p and 307 labels for q. The regularization parameters λ_p and λ_q for p and q respectively were manually adjusted to 0.01 and 0.01. The value of T of the truncated linear prior [See Eq. (3)] was chosen to be 0.8 for both p and q fields.

Fig. 1(a) shows the observed low resolution Vase image with light source position $(0,0,1)$ and of size 64×64. The Fig. 1(b-c) shows the images of size 128×128, reconstructed using bi-cubic interpolation of p, q and ρ fields and super-resolved image using the proposed method respectively. Although perceptually the images (b) and (c) look similar, the mean square error (MSE) comparison (discussed later) shows that graph cuts based approach is indeed better. Fig. 2(a) shows the high resolution ground truth for depth of Vase image. The Fig. 2(b-c) shows the up-sampled depth reconstructed using bi-cubic interpolation of p and q fields and super-resolved depth using the proposed method respectively. Perceptually the depth map obtained by using bi-cubic interpolation looks better than that obtained using the proposed method. However, by fine tuning the regularization parameter λ_x and the threshold T_x, where $x = p, q$, it is possible to get better results.

Next we consider a real object Jodu. Eight images of Jodu were captured with different light source positions. We consider the actual observed Jodu images of size 234×234 as the desired high resolution images. These images are decimated to obtain low resolution images of size 117×117, which now become the given low resolution observations.



(a) (b) (c)

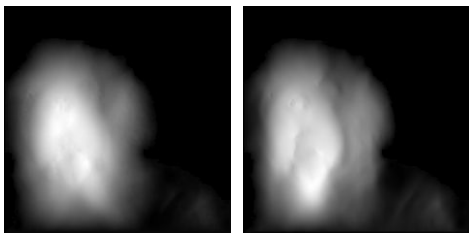
Figure 2: Depth map for Vase Image (a) Ground Truth (b) Up-sampled depth reconstructed using bi-cubic interpolation of p, q and ρ fields (c) Super-resolved using the proposed approach

(a) (b) (c)

Figure 3: (a) Observed image with light source position $(0.8389, 0.7193, 1)$ (b) Up-sampled image reconstructed using bi-cubic interpolation of p, q and ρ fields (c) Super-resolved Jodu image using proposed approach

For the Jodu image, we observed that the initial values of p lie in the range $(-1, 1)$ and that of q lie in the range $(-0.6, 0.6)$. Hence, depending on the frequency distributions of the respective entities, we use 440 labels for p and 420 labels for q. The regularization parameters λ_p and λ_q for p and q respectively were manually adjusted to 0.008 and 0.0259. The value of T of the truncated linear prior [See Eq. (3)] was chosen to be 0.175 for both p and q fields.

Fig. 3(a) shows one of the observed low resolution Jodu image of size 117×117. Fig. 3(b) shows the high resolution images of size 234×234 reconstructed from the bi-cubic interpolation of the p, q and ρ fields for the same light source positions. The super-resolved images using the proposed approach is shown in Fig. 3(c). Although visually there is not much difference in the super-resolved images reconstructed using graph cuts and bi-cubic interpolation, our quantitative analysis (discussed later) shows that the images reconstructed using graph cuts are indeed superior. The depth maps reconstructed using bi-cubic interpolation of p, q and ρ and that obtained using the proposed approach for the Jodu image are shown in Fig. 4(a-b). It may be noted here that we do not have the true depth map for comparison since the laser scanner does not work well with objects with discontinuities. One can observe from the Fig. 4(b) that discontinuities in depth



(a) (b)

Figure 4: Depth map for Jodu Image (a) Up-sampled depth reconstructed using bi-cubic interpolation p, q and ρ fields (b) Super-resolved using the proposed method

are much better revealed as compared to Fig. 4(a) that was reconstructed using bi-cubic interpolation.

For quantitative comparison, we use mean square error (MSE) as a figure of merit. Table 1 shows the MSE comparison for the super-resolved image and the depth map (for both Vase and Jodu images) and the case when interpolated values of the surface gradients and albedo are used for reconstruction of the up-sampled depth and intensity map. Although, not much difference can be seen in the high resolution images reconstructed using the two methods, the MSE values clearly show that the high resolution images obtained using our graph cuts based approach are much better than those obtained using bi-cubic interpolation. Due to the use of edge preserving smoothness term, the reconstructed image using graph cuts minimization is closer to the actual high resolution images. The high resolution depth obtained for the Vase image using our approach shows a superior MSE performance as compared to bi-cubic interpolation. It may be mentioned here that we do not have the actual depth map for Jodu, we use the depth map obtained using the actual observed 234×234 images with photometric stereo as the reference depth map for calculating MSE. Since the reference depth map is not the actual depth map (with edges properly defined), the MSE performance when depth is obtained using the proposed approach is poorer when compared to depth obtained using bi-cubic interpolation.

We now discuss the time complexity of our algorithm. The graph cuts based super-resolution approach takes around $5-7$ minutes for convergence (on a 1.33 GHz processor for 234×234 image size) while it takes hours for convergence when simulated annealing with edge preservation is used [8]. In [8] the authors mention that the time for convergence using simulated annealing is of the order of hours. Our approach, on the other hand, takes few minutes. It may be mentioned that although we are not using the other constraints used in [8] while optimization since the time required for simulated annealing is much larger as the cost is computed by changing the label of a single pixel in each move. On the other hand, in graph cuts the labels of a number of pixels get changed together in each move. One can thus observe the kind of complexity reduction that has been achieved through the graph cuts based formulation for super-resolution. Hence, our approach performs much better when compared to computationally expensive optimization methods. It may also be mentioned here that in [8] the discontinuity preservation prior terms consisted of edge preserving line fields. However, we use a truncated absolute distance for edge preservation. Hence, we do not compare our results with the simulated

Table 1: MSE comparison for the high resolution Vase and Jodu images and depth map obtained using bi-cubic interpolation and our super-resolution approach with an upsampling factor of 2 with different source positions. The (DEPTH) row in the table gives the MSE for the depth field.

Source position for Vase Image	MSE	
	Bi-cubic Interpolation	Graph cuts
(0, 0, 1)	86.03	14.82
(DEPTH)*	6.71	1.68
For Jodu Image		
(0.8389, 0.7193, 1)	240.13	43.55
(-0.1763, -0.5596, 1)	544.01	51.00
(DEPTH)	9.57	68.79

* Only the center portion of the Vase is used for MSE calculation.

annealing based super-resolution method proposed in [8].

7 Conclusion

In this paper, we used graph cuts optimization for obtaining a super-resolved depth map and intensity map using photometric cue. The surface gradients were modeled as separate MRFs. We used a smoothness prior with discontinuity preservation. The results show that the super-resolved image and depth obtained using our approach reveal edges better than the up-sampled depth and images obtained using general interpolation techniques. The quantitative measure (MSE) also shows that the graph cuts based super-resolution scheme is superior than these methods. Also, our approach takes a few minutes for convergence which is very much less than the super-resolution scheme that uses simulated annealing for optimization [8] (takes hours for convergence). It can be seen from the results that our graph cuts based super-resolution approach provides a time-effective method for super-resolution which is very much required in a practical scenario.

8 Acknowledgements

We are thankful to Dr. André Jalobeanu, LSIIT, Université de Louis Pasteur, Strasbourg, France for his constructive suggestions and comments.

References

[1] Tikhonov A. N. and Arsenin V. Y. *Solution of Illposed Problems*. W.H. Winston, Washington D.C., 1977.

[2] Rajan D. and Chaudhuri S. Generalized interpolation and its application in super-resolution imaging. *Image and Vision Computing*, 19(13):957–969, 2001.

[3] Shekarforoush H., Berthod M., Zerubia J., and Verman M. Sub-pixel bayesian estimation of albedo and height. *International Journal of Computer Vision*, 19(3):289–300, 1996.

[4] Ur H. and Gross D. Improved resolution from sub-pixel shifted pictures. *CVGIP: Graph, Models and Image Process.*, 54, 1992.

[5] Berthod M., Shekarforoush H., Verman M., and Zerubia J. Reconstruction of high resolution 3d visual information. *Technical Report, RR-2142, INRIA*, 1993.

[6] Irani M. and Peleg S. Improved resolution by image registration. *CVGIP: Graph, Models and Image Process.*, 53, 1991.

[7] Ng M. K., Koo J., and Bose N. K. Constrained total least squares computation for high resolution image reconstruction with mulisensors. *International Journal of Systems and Technologies*, 12, 2002.

[8] Joshi M. V. and Chaudhuri S. Simultaneous estimation of super-resolved depth map and intensity field using photometric cue. *Computer Vision and Image understanding*, 101:31–44, 2006.

[9] Cheeseman P., Kanefsky B., Hanson R., and Stutz J. Super-resolved surface reconstruction from multiple images. *Technical Report, FIA-94-12, NASA Ames Research center, Artificial Intelligence Branch*, 1994.

[10] Tan P., Lin S., and Quan L. Resolution-enhanced photometric stereo. *European Conference on Computer Vision*, 3:58–71, 2006.

[11] Schultz R. R. and Stevenson R. L. A bayesian approach to image expansion for improved definition. *IEEE Transactions on Image Processing*, 3(3):233–242, 1994.

[12] Tsai R. Y. and Huang T. S. Multiframe image restoration and registration. *Advances in Computer Vision and Image Processing, JAI Press*, 1984.

[13] Wu T. P. and Tang C. K. Dense photometric stereo using a mirror sphere and graph cut. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.

[14] Mudenagudi U., Singla R., Kalra P., and Banerjee S. Super-resolution using graph-cuts. *Asian Conference on Computer Vision*, 2006.

[15] Kolmogorov V. and Zabih R. Multi-camera scene reconstruction via graph cuts. *European Conference on Computer Vision*, 3:82–96, 2002.

[16] Kolmogorov V. and Zabih R. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26, 2004.

[17] Boykov Y., Veksler O., and Zabih R. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:1222–1239, 2001.

[18] Boykov Y. and Kolmogorov V. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004.

Multi-scale Adaptive Mask 3D Rigid Registration of Ultrasound and CT Images

Zhijun Zhang, Jerome Schmid, Man Kwan Soo,
Bailly Yan, Chung Kwong Yeung
Surgery Department
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong
alexzhang@surgery.cuhk.edu.hk

Abstract

3D Rigid intra-operative image registration is an important technique to provide guidance for pre-operative information from different image modalities. Due to the artefacts that cause correspondence ambiguity, accurate registration of US with other image modalities such as computed tomography (CT) is still a challenging problem. We propose a method which considers the registration problem of US and CT images as a multi-scale regional information saliency and local similarity selection process. We design our method as a multi-stage approach in which global and local rigid registrations alternate in each stage. During the local registration, the US image regions with high feature saliency and similarity with CT image will be selected and joined as region masks for the next global registration, other parts will not be considered in order to remove the correspondence ambiguity. The performances of our method are compared with a typical intensity based registration method on phantom and real patient images.

1 Introduction

3D Ultrasound (US) imaging is widely used in image guided surgery due to its non-ionizing effect, low cost and real-time properties. However, US images are spoiled by speckle noise and artefacts [3]. The artefacts regions do not correspond to meaningful anatomical structure information. This information incompleteness phenomenon often makes the comprehension of anatomical structure very difficult. Registration between the US images with a complementary modality, such as computed tomography (CT) or magnetic resonance imaging (MRI), appears as a promising solution to improve US image understanding. By correctly aligning the CT images with the US images, all of the pre-operative information extracted from the CT (e.g. segmentation of organs, major vessels and pathologies detections) can be augmented on the US image [2, 7]. This technique will bring great convenience and improve the efficiency and safety during the surgical practice. Accurate registration between the CT and US images is the critical problem in ultrasound augmented reality.

The multi-modal image registration can be categorized into model based, feature based and voxel based registration methods. Among these methods, voxel based methods use directly the intensity information to match the source and target images, it is not necessary to segment the images or extract models from images. They are very suitable for computer implementation. There are several intensity based multi-modal image registration methods [9, 13, 11, 15] and they work well on registrations between CT and MR images. Maes *et al* [9] proposed mutual information (MI) based registration method, Studholme *et al* [13] proposed an improvement of overlap independent method by using normalized mutual information (NMI). Roche *et al* [11] proposed a method on correlation ratio (CR) and Woods *et al* [15] proposed a method on partition intensity ratio (PIU). In these methods, image intensities are considered as random variables with identical independent distributions. The image similarity functions will measure the dependency or correlation between the two random variables. It can be represented as function of the joint probability density function (PDF) which is evaluated from corresponding intensity pairs. The correspondence between artefacts regions of US image and CT image will add the unreliable intensity pairs into the joint PDF and bias it. The influence of US artefacts in registration has been reported in [5]: the result will not always correspond to the global optimum or even to a local optimum.

A solution is to detect the useful information for registration. By extracting the regions where the images have better correspondence information, the performance of the intensity based methods can be greatly improved. Huang *et al* [5] used a threshold to extract the useful voxels in the US. However, this simple operation does not work properly on US image with complex anatomical structures such as abdomen and brain. Roche *et al* [11] proposed a robust estimation of bivariate function together with a correlation ratio method to suppress the correspondence outliers between MRI and US. The results are usually dependent on the parameters tuning and the Powell's optimization process is time consuming. Penney *et al* [10] extracted a vessel probability density map from the US images and used it to register with the MRI images. This method needs a learning process by using a large amount of US images together with a manually determined threshold for MRI images. Leroy *et al* [8] and Wein *et al* [14] used noise models to detect the artefacts regions. In real application, it is difficult to find a general model for artefacts to achieve good performances. Local features can provide unique and reliable information for registering the images with less trustable information. Stewart *et al* [12] proposed a method to register the retinal images by using local features. The registration starts from the most accurate local feature matching and then propagates with more global feature matching. Salient regions have been used as features for registration recently because of its higher robustness. Huang *et al* [1] has used multiple salient regions for 2D image registrations.

We propose a multi-scale 3D adaptive mask MI based CT to US images registration method. It is an iterative method with several stages. In each stage a global registration and some local block matchings are carried out. Regions with low saliency or low similarity with CT image will be excluded from registration, they usually coincide with the artefact image regions. The global and local registration process will alternate until the whole registration converges.

The rest of the paper is organized as follows. Our novel intensity based registration method is explained in the Section 2. The experiments and datasets are introduced in Section 3. Results and discussion are shown in Section 4. Finally, conclusions are given in Section 5.

2 Methods

2.1 Matching Ambiguity in Intensity Based Registration Methods

Due to the artefacts in the US images, the correspondence between the US and CT images may have ambiguity. MI based method prefers the transform which brings the joint histogram to be more clustered.This problem will happen to all of the information theory based registration method including NMI method. Fig.1 shows a negative MI metric of a 3D phantom US and CT images with different masks of the US image. The images in left column are axial slices of a CT image and the overlaid US and CT images which are already registered. We can see in the right part of the US image, a region with shadow artefacts exists. Images in middle column shows two different mask images of US volume used for registration, the upper one is the fan shape mask and the lower one is the vessel area mask. The figures in right column show negative MI metrics with different US image masks as functions of translational parameters errors in x and y directions. We can see the the similarity metric with fan mask does not correspond with a global optimum while that with a vessel area mask shows a global minimum in the correct transform parameters.

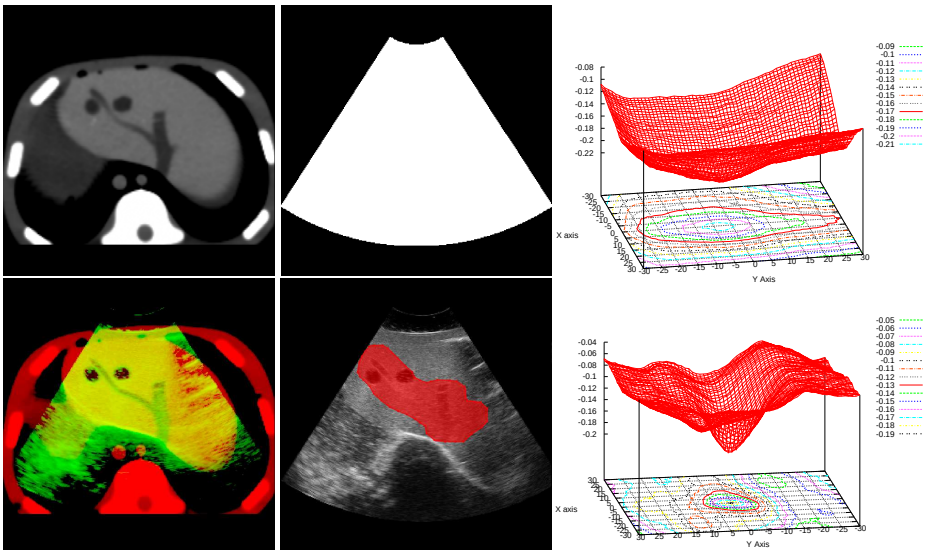


Figure 1: Left column: one axial slice of the CT image and the color overlaid US and CT corresponding axial slice; Middle column: the fan shape mask and the vessel mask of US image; Right column: the negative MI metrics plotted with fan shape mask and the vessel mask against the registration parameters error around the ground truth parameters.

2.2 Combination of Global and Local Registrations

We design our method as a combined global and local rigid registration method. This method combines the advantage of robustness for global registration and accuracy for local registration. We use MI for global and local similarity metric in our implementation

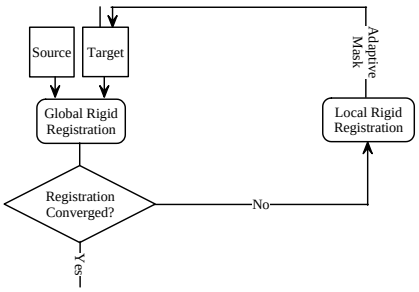


Figure 2: The combination of global and local rigid registrations.

but any multi-modal intensity based similarity metric can be embedded into our registration framework. We begin our registration using global MI registration method with whole US fan mask. We then divide our image into uniform blocks and analyze the local region saliency and their similarities with CT image. Blocks with high local saliency and high similarity with CT will be selected for next global registration. The selected local region will be joined and used as the mask for global registration. The whole registration process will alternate between global and local registrations until the it converges. Fig.2 shows the collaboration of the global and local registration methods.

2.3 3D Region Saliency Calculation

The region with useful information and the artefacts can be extracted by using the concept of region saliency [6]. Saliency is the measurement of an unpredictability of local attributes over a scale. It is proposed for general images and it is also suitable for medical images. The local attributes can be intensity values or colors. The scale can be considered as a sphere with a certain radius. Larger saliency measurement means bigger unpredictability and probability density function magnitude change over scale, so intuitively it means a bigger dissimilarity over scale. Higher saliency regions will be less possible to be the area of artefacts since in these areas when changing the scale, the intensity is usually a reordering and the difference of PDF magnitude is very small.

The region saliency detection consists of three steps as described in [6]. The local saliency will be the product of local entropy $H_D(s,\mathbf{x})$ and local probability density difference $W_D(s,\mathbf{x})$ at the optimum scale:

$$S_D(s_p,\mathbf{x}_0) = H_D(s_p,\mathbf{x}_0)W_D(s_p,\mathbf{x}_0). \tag{1}$$

In continuous case, the entropy will be described by:

$$H_D(s,x_0) = -\int p(i,s,\mathbf{x}_0) log p(i,s,\mathbf{x}_0) di, \tag{2}$$

with i the intensity index, $p(i,s,\mathbf{x}_0)$ the Parzen window estimation of the probability density function around $\mathbf{x}_0$ with a scale s. We denote the Parzen window local PDF estimation as in [4]:

$$p(i,\sigma_s,\mathbf{x}_0) = \frac{1}{|\Omega|}\int_{\mathbf{x}\in\Omega} g_\sigma(i-I(\mathbf{x}))g_{\sigma_s}(\mathbf{x}-\mathbf{x}_0)d\mathbf{x}, \tag{3}$$

with g_σ a Gaussian kernel function and $g_{\sigma_s}(\mathbf{x}-\mathbf{x}_0)$ a Gaussian function weighting each of the intensity Gaussian value according to the distance of the points to the region center $\mathbf{x}_0$, Ω is the local image area and $|\Omega|$ is the volume size of these region. Change of σ_s will changes the scale of the local joint entropy, so we directly use σ_s as the scale, that is $\sigma_s = s$.

The optimal local scale s_p is determined by:

$$s_p = \{s : \frac{\partial H_D(s,\mathbf{x}_0)}{\partial s} = 0, \frac{\partial^2 H_D(s,\mathbf{x}_0)}{\partial s^2} < 0\}, \tag{4}$$

and the continuous partial derivative of $H_D(s,\mathbf{x}_0)$ with respect to scale will be:

$$\frac{\partial H_D}{\partial s} = -\int\int (logp(i,s,\mathbf{x}_0)+1) g_\sigma(i-I(\mathbf{x})) g_s(\mathbf{x}-\mathbf{x}_0)(\frac{\mathbf{x}-\mathbf{x}_0}{s})^2 did\mathbf{x}. \tag{5}$$

Then the optimal local scale can be obtained by solving nonlinear equation (4) by substituting $\frac{\partial H_D}{\partial s}$ by (5). The inter-scale saliency measure, $W_D(s,\mathbf{x}_0)$ is defined by:

$$W_D(s,\mathbf{x}_0) = s\int |\frac{\partial}{\partial s} p(i,s,\mathbf{x}_0)| di \tag{6}$$

When the optimal scale of a point is obtained, we can substitute s_p into equation (6) and (1) to obtain the saliency measurement around point $\mathbf{x}_0$.

Instead of evaluate the saliency at each of the voxel, we divide the 3D US image into uniform blocks, then we evaluate the saliency at the block center. We assume the saliency measurement is continuous and smooth, the saliency evaluated at the block center represents the block saliency. This will decrease the computing time distinctly. At each of block center, we obtain the optimum scale for local entropy and the region saliency. We reorder the blocks by their center points saliency from high to low. We choose a portion (we use 70%) of the blocks with high saliency value for local block matching.

2.4 Polar Coordinate Image Processing

Because the beam ray characteristic of US imaging, the image is actually sampled along the ultrasonic beams. The artefacts are also distributed along the beam rays. When we evaluate the entropy of the US image or joint entropy of US and CT images, it is more reasonable to sample the US image along the beam rays. The beam rays form a coordinate space of Polar coordinate space. So instead of measuring the entropy and MI by sampling uniformly in Cartesian coordinate space, we sample the US images in Polar coordinate space. A typical 3D Cartesian and Polar coordinate space for US is shown in fig.3 in the top row. The coordinate transform information can be easily obtained from the geometrical information of the 3D US volume. An example of the same number of sampled points for MI evaluations in the Cartesian and Polar coordinates are shown in the bottom row of fig.3 from left to right. The US image is used as source image and the sample points are randomly chosen in it. We can see the uniform distributions in Cartesian and Polar coordinate space respectively in these two methods. We can see obviously that the sample points are uniformly distributed along the ultrasonic beam rays instead of in the 3D rectangular space.

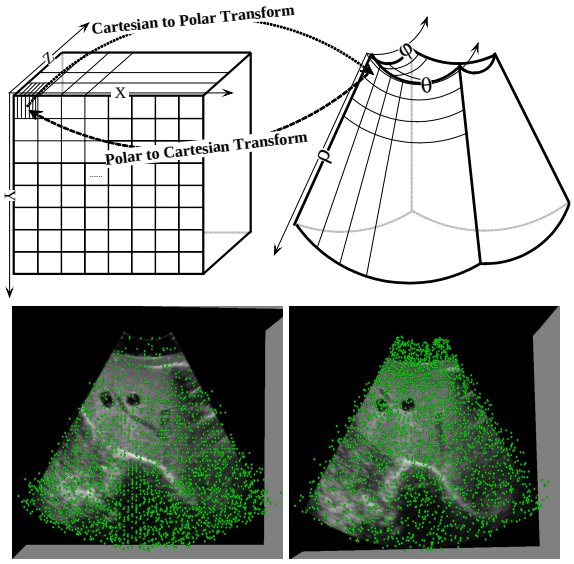


Figure 3: 3D Cartesian coordinate and Polar coordinate spaces. The upper row shows the two coordinates. The bottom row shows the Cartesian coordinate sampling scheme and Polar coordinate scheme in MI calculating. Each sample point is represented by a green dot.

2.5 Registration by Dominant Block Matching

After the blocks with high saliency value are detected, we will use these blocks to locally recover the rigid transform between the US and the CT images. For each of the blocks, an MI based registration is used to acquire the local rigid transform. The region is defined as a box in 3D Polar coordinate. The side length of the region is equal to the size of the optimum scale at the block center. The region with the optimal saliency scale as the radius is used for the registration. The US sub-region is sampled in 3D Polar coordinate and the MI similarity metric is optimized. When all of the local block matchings are finished, we rearrange these blocks by the final MI value. We again take the portion (70%) of the blocks with higher local similarity measurement as useful blocks for next stage. The local rigid transform parameters obtained from these selected blocks will be averaged and it will be used for the initialization of the next global registration.

3 Data and Experiments

3.1 Data Acquisition

We used both *in vitro* and *in vivo* datasets to evaluate our method together with MI method. The former came from a multi-modal abdominal phantom (model 057, CIRS Inc.®), and the latter were the abdominal images of a patient. All 3D US images were taken from a GE Voluson® 730 machine with a 3D transducer of model RAB2-5L. The CT images were taken from a Helical CT machine of GE system®. The images were taken while the

patients held their breath. The CT and US image characteristics for *in vitro* and *in-vivo* experiments including image dimension, voxel size and number of datasets are shown in Table 1.

Experiments	Data	Image Information		
		Dimension	Voxel size(*mm*)	Numbers
in vitro	US phantom	$256 \times 256 \times 256$	$0.915 \times 0.708 \times 0.580$	3
	CT phantom	$256 \times 256 \times 119$	$1.25 \times 1.25 \times 1.25$	1
in vivo	US patient	$256 \times 256 \times 256$	$0.840 \times 0.591 \times 0.640$	3
	CT patient	$512 \times 512 \times 177$	$0.625 \times 0.625 \times 1.25$	1

Table 1: The *in vitro* and *in vivo* US and CT dataset characteristics.

3.2 Experiments

The registration results are evaluated from both visual inspection and quantitative experiments. We evaluate the registration method by starting the registration with random initial parameters for multiple times. For each pair of the images to be registered, a ground truth rigid transform is obtained by using a feature point initialized and intensity based registration software. Several pairs of corresponding points are manually picked by a radiologist and an intensity based method with a manually labeled ROI will refine the initial result. We represent the 3D rigid transform by using six parameters, three for rotations and three for translations. For each datasets, we evaluated these parameters by running 100 registrations, each of which was initialized with an arbitrary transform. Each of the parameters was generated by adding an arbitrary displacement error of parameters to the ground truth parameters. In generating the arbitrary displacement error for parameters, each component of the parameters displacement was chosen within an error range. In our tests, the ranges for the translational and rotational error components were $\pm 30mm$, and $\pm 0.349rad$ respectively.

4 Results and Discussion

4.1 Accuracy

We list the accuracy test results in Table.2. For phantom and patient registration results, the parameters errors from multiple datasets are averaged. In both the phantom and the patient experiments, the parameter components errors with the ground truth parameters are quite high after MI based registration, while after our proposed method, the component errors with ground truth are much decreased.

4.2 Qualitative Evaluation

Registration of a US and CT phantom image is shown in shown in fig. 4. The first two images in upper row are the CT and US image respectively. The first two images in bottom row are the registration results by using MI registration method and our proposed method. The registration results are shown by one axial slice of color overlaid images. We can see there is small misalignment after the MI registration while after registration by using our method, the resampled US image overlays with the CT image much better. We use three stages in this experiment and figure in upper right shows the adaptively selected

Experiments	Methods	Parameters Errors							
		Rotation(*rad*)				Translation(*mm*)			
		ΔR_x	ΔR_y	ΔR_z	$\|\|\Delta R\|\|$	ΔT_x	ΔT_y	ΔT_z	$\|\|\Delta T\|\|$
US to CT phantom	MI	0.035	0.05	0.033	0.07	1.60	6.52	1.77	6.94
	AMMI	0.033	0.029	0.027	0.06	0.78	0.83	0.92	1.46
US to CT patient	MI	0.05	0.04	0.06	0.08	0.65	5.17	1.33	6.64
	AMMI	0.03	0.04	0.03	0.05	0.49	0.96	1.24	1.65

Table 2: The results of MI and our proposed registration method for the random initial parameters tests. MI: mutual information registration method; AMMI: our proposed adaptive mask mutual information registration method.

mask. We can see the most of shadow region in the right part of the image is excluded. The bottom right figure shows the negative MI metric with adaptive mask plotted against the translational parameters errors around the alignment. The metric function shows an unique and accurate optimum at the matching parameters and the metric function is quite smooth. Registration of a real patient is shown in fig.5. The top row shows the CT and US

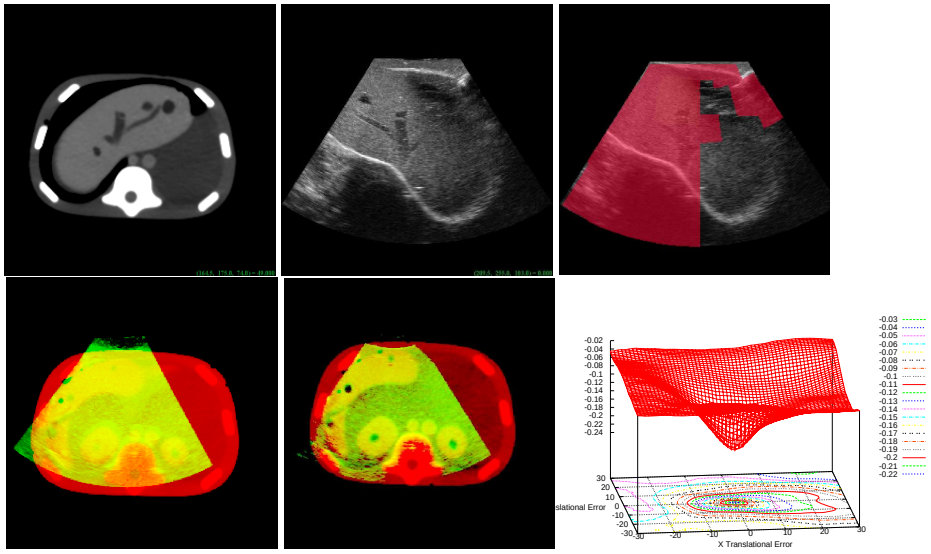


Figure 4: Phantom US to CT registration results by using MI and our method.

images of a patient liver. In the bottom row, from left to right are the registration results by using MI and our method shown by color overlaid . We can see the improvement of alignment near the inferior vena cava.

5 Conclusion

In this paper, we have presented a new rigid 3D US to CT image registration method. We have adaptively selected the regions with high saliency and similarity with CT image

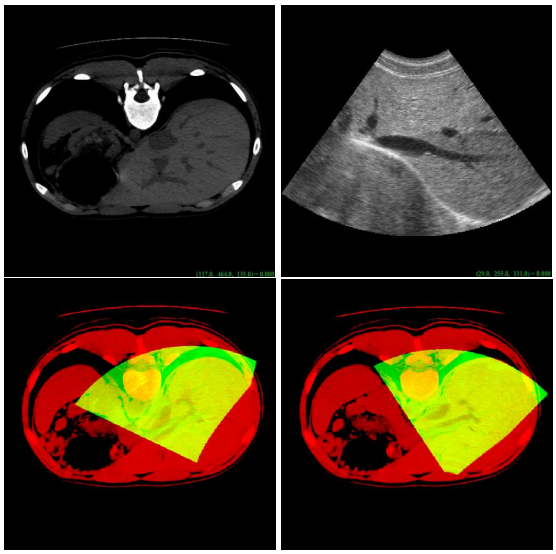


Figure 5: Registration results of US to CT patient liver by using MI and our proposed method.

as useful information for registration. We compared our method with a typical intensity based multi-modal registration method — MI based method, the results of phantom and real patient datasets show the improvement of the accuracy of the registration parameters. This method can be applied to the applications where only partial image exists for registration.

6 Acknowledgement

The patient datatsets used in this paper were provided by Dr. Winnie Chu and Leung Yee Fong, Vivian in Prince Wales Hospital of Hong Kong. This research work is supported by grant from the Jockey Club Minimally Invasive Surgical Skills Centre (MISSC) of Hong Kong.

References

[1] Hybrid image registration based on configural matching of scale-invariant salient region features. In *Second IEEE Workshop on Image and Video Registration*, pages 167–176, 2004.

[2] S.R. Aylward, J. Jomier, J.P. Guyon, and S. Weeks. Intra-operative 3d ultrasound augmentation. In *IEEE International Symposium on Biomedical Imaging*, pages 421–424, 2002.

[3] B. Block. *The Practice of Ultrasound – A Step by Step Guide to Abdominal Scanning*. 2004.

[4] G. Hermosillo and O. Faugeras. Variational methods for multimodal image matching. *International Journal of Computer Vision*, 50(3):329–343, 2001.

[5] X.S. Huang, N.A Hill, J. Ren, and T.M. Peters. Dynamic 3d ultrasound and mr image registration of the beating heart. In *Medical Image Computing and Computer Assisted Intervention , MICCAI 2005*, volume 3750 of *Lecture Notes in Computer Science*, pages 171–178, 2005.

[6] T. Kadir, A. Zissereman, and M. Brady. An affine invariant salient region detector. In *European Conference on Computer Vision - ECCV 2004*, volume 3021 of *Lecture Notes in Computer Science*, pages 228–241. Springer Berlin, 2004.

[7] T. Lange, S. Eulenstein, M. Hunerbein, H. Lamecker, and P.M. Schlag. Augmenting intraoperative 3d ultrasound with preoperative models for navigation in liver surgery. In *Medical Image Computing and Computer Assisted Intervention , MICCAI 2005*, volume 3217 of *Lecture Notes in Computer Science*. Springer, Sep 2004.

[8] A. Leroy, P. Mozer, Y. Payan, and J. Troccaz. Rigid registration of freehand 3d ultrasound and ct-scan kidney images. In *Medical Image Computing and Computer Assisted Intervention , MICCAI 2005*, volume 3216 of *Lecture Notes in Computer Science*, pages 837–844. Springer, Sep 2004.

[9] F. Maes, A. Collignon, and D. Vandermeulen. Multimodality image registration by maximization of mutual information. *IEEE Trans. on Med. Imag.*, 16(2):181–187, 1997.

[10] G.P. Penney, J.M. Blackall, M.S. Hamady, T. Sabharwal, A. Adam, and D.J. Hawkes. Registration of freehand 3d ultrasound and magnetic resonance liver images. *Medical Image Analysis*, 8(1):81–91, 2004.

[11] A. Roche, G. Malandain, N. Ayache, and X. Pennec. Multimodal image registration by maximization of the correlation ratio. *INRIA Research Report*, (3378), 1998.

[12] C.V. Stewart, C.L. Tsai, and B. Roysam. The dual-bootstrap iterative closest point algorithm with application to retinal image registration. *IEEE Transaction on Medical Imaging*, 22(11):1379–1394, 2003.

[13] C. Studholme, D.J. Hawkes, and D.L.G. Hill. An overlap invariant entropy measure of 3d medical image alignment. *Pattern Recognition*, 32(11):71–86, 1999.

[14] W. Wein, R. Barbara, and N. Nassir. Automatic registration and fusion of ultrasound with ct for radiotherapy. In *Medical Image Computing and Computer Assisted Intervention , MICCAI 2005*, Lecture Notes in Computer Science, pages 303–311. Springer, Oct 2005.

[15] R. P. Woods, J. C. Mazziotta, and S. R. Cherry. Mri-pet registration with automated algorithm. *Journal of Computer Assistant Tomography*, 17(4):536–546, 1993.

Geometrical constraint based 3D reconstruction using implicit coplanarities

Ryo Furukawa
Faculty of information sciences, Hiroshima City University, Japan
ryo-f@cs.hiroshima-cu.ac.jp

Hiroshi Kawasaki
Faculty of engineering, Saitama University, Japan
kawasaki@cgv.ics.saitama-u.ac.jp

Abstract

Coplanarity is a relationship of a set of points that exist on a single plane. Coplanarities can be easily observed in a scene with planer surfaces, and these types of coplanarities have been widely used for 3D reconstructions based on geometrical constraints. Other types of coplanarities that can be observed from images are those observed as cross sections of planes and scenes; for example, points lit by a line laser, or boundary points of a shadow of a straight edge. Although these types of coplanarities have been implicitly used in variations of light sectioning methods, they have not been used in an unified manner with the former types. In this paper, we describe a new 3D reconstruction method based on coplanarities and other geometrical constraints. In particular, we make use of the above two types of coplanarities in an unified manner. This enables us to reconstruct 3D scenes scanned using line lasers or shadows of straight edges observed by a partially-calibrated single camera utilizing geometrical relationships between the planes in the scenes and the planes of line lasers or the planes of shadow boundaries.

1 Introduction

If a set of points exist on a plane, they are said to be coplanar. For example, if a scene includes a planer surface, points on the surface are coplanar. A scene composed of plane structures has many coplanarities.

On the other hand, there are other types of coplanarities. In a 3D space, there exist an infinite number of coplanarities that are not explicitly observed in ordinary situations, but could be observed under specific conditions. For example, points lit by a line laser are coplanar points. Another example is a set of points on a boundary of a cast-shadow of a straight edge. These types of coplanarities are not visible until the lasers or the shadows are cast on the scene. Let us call the former types of coplanarities as *explicit coplanarities* since they can be observed as visible surfaces of the scene, and let us call the latter types as *implicit coplanarities*.

Explicit coplanarities can be observed in scenes composed of planer surfaces, and have been widely used as geometrical constraints for 3D reconstructions [11, 9, 1, 10, 7, 6].

In vision researches, implicit coplanarities have been used in variations of light sectioning methods. In most of these researches, the light planes are first calibrated using some kind of calibration objects [3, 4, 5] and then the points on the laser planes are reconstructed using triangulation. In these researches, implicit coplanarities were not treated as geometrical constraints that can be solved by themselves to reconstruct 3D structures.

In this paper, we describe a new 3D reconstruction method based on coplanarities and other geometrical constraints. In our method, we use both two types of coplanarities in an unified way to reconstruct projective 3D information. This enables us to reconstruct projective 3D scenes with curved surfaces by using implicit coplanarities obtained by scanning the scenes with line lasers, or shadows of straight edges, observed by a partially-calibrated single camera.

Although coplanarities play an important role for shape reconstruction, it is known that the the geometrical constraints other than coplanarities (such as orthogonalities or parallelisms) are needed to achieve Euclidean reconstructions of the scenes[11]. Because of the unified treatment of both the implicit and explicit coplanarities, we can use geometrical constraints between both types of planes without discriminating them to achieve Euclidean reconstructions. This widens the applicabilities of our method. For example, a scene with curved surfaces can be densely reconstructed either by scanning the scene with a projector composed of two line lasers and utilizing geometrical constraints between the line lasers, or by scanning the scene with a single line-laser projector and utilizing geometrical constraints found in the scene (such as orthogonalities of the surfaces of the objects).

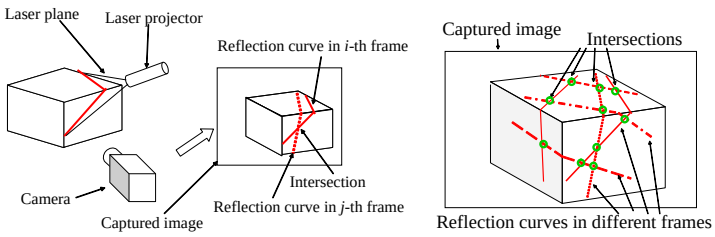
2 Related studies

Explicit coplanarities have been used in analysis of line drawings or 3D reconstructions based on geometrical constraints [11, 9, 1, 10, 7, 6]. In those studies, only scenes with planer surfaces are targeted, because they use only visible coplanarities and geometrical constraints that exist for those planer surfaces.

In computer vision researches, implicit coplanarities have been used, although unconsciously, in light sectioning methods. Recently, several researchers developed handheld 3D scanners based on light sectioning methods [3, 4, 5]. In these methods, the laser planes are calibrated by using calibration objects such as fixed frames, markers, or known planes, then the points on the laser (shadow) planes are reconstructed using triangulations. Bouguet *et al.* proposed a method in which the scene is scanned by shadows of a straight edge to reconstruct the scene [2]. Their technique requires calibration of camera parameters, a light source position, and a reference plane. Implicit coplanarities in these works are only planes for triangulations, and they should be calibrated first by using some calibration objects(known frames, markers, or planes). In contrast to these methods, our method does not require any special calibration objects.

3 Shape reconstruction from coplanarities

Reconstruction in the proposed method is realized by solving the simultaneous equations constructed from both the coplanarities and the metric constraints. As described later, metric constraints are formulated with nonlinear equations, whereas coplanarity



(a) (b)

Figure 1: (a)An example configuration of the system. (b)Points of intersections.

constraints can be described by linear equations. Therefore, our method first solves linear simultaneous equations achieving projective reconstruction, and upgrades the solution to the Euclidean space.

3.1 Projective reconstruction

An example of our system consists of a camera and a line laser projector, as shown in Figure 1(a). The focal length of the camera may be unknown. The line laser beam from the projector is reflected at the surfaces of the scene and detected by the camera. These points are implicit-coplanar. A scanning process is performed by capturing a sequence of images with the camera while moving the projector back and forth. Scanning can also be performed by moving a cast shadow of a straight edge over the scene. Multiple reflection curves are obtained from the image sequence since they move in the image with the motion of the projector. The problem to be solved is the estimation of the positions of the projected laser planes from the observed implicit coplanarities. By drawing all the reflections in different frames in a image, those curves have intersections (Figure 1(b)). We can obtain geometrical constraints of coplanarities from these intersections since each of those points exists on multiple planes.

Suppose a set of N planes including both implicit and explicit planes. Let j-th plane of the set be π_j. We express the plane π_j by the form

$$a_j x + b_j y + c_j z + 1 = 0 \tag{1}$$

in the camera coordinates system.

Suppose a set of points such that each point of the set exists on intersections of multiple planes. Let the i-th element of the set be represented as ξ_i and exist on the intersection of π_j and π_k. Let the coordinates (u_i, v_i) be the location of the projection of ξ_i onto the image plane. We represent the camera intrinsic parameter by $\alpha = p/f$, where f is the focal length and p is the size of the pixel. We define $a_j^* = \alpha a_j$ and $b_j^* = \alpha b_j$. The direction vector of the line of sight from the camera to the point ξ_i is $(\alpha u_i, \alpha v_i, -1)$. Thus,

$$a_j(-\alpha u_i z_i) + b_j(-\alpha v_i z_i) + c_j(z_i) + 1 = 0, \tag{2}$$

where z_i is the z-coordinate of ξ_i. By dividing the form by z_i and using the substitutions of $t_i = 1/z_i$, $a_j^* = \alpha a_j$, and $b_j^* = \alpha b_j$, we get

$$-(\alpha u_i)a_j^* - (\alpha v_i)b_j^* + c_j + t_i = 0. \tag{3}$$

Since ξ_i is also on π_k,

$$-(\alpha u_i)a_k^* - (\alpha v_i)b_k^* + c_k + t_i = 0. \tag{4}$$

From the forms (3) and (4), the following simultaneous equations with variables a_j^*,b_j^*,c_j,a_k^*,b_k^* and c_k can be obtained:

$$-u_i a_j^* + u_i a_k^* - v_i b_j^* + v_i b_k^* + c_j - c_k = 0 \tag{5}$$

We define $\mathbf{L}$ as the $M \times 3N$ coefficient matrix of the above simultaneous equations, and $\mathbf{x} = (a_0^*, b_0^*, c_0, a_1^*, b_1^*, c_1, \cdots, a_{N-1}^*, b_{N-1}^*, c_{N-1})^\top$ as the solution vector for all the M intersections and the N planes. Then, the equations can be described by a matrix form as

$$\mathbf{L}\mathbf{x} = \mathbf{0}. \tag{6}$$

Simultaneous equations of forms (5) have trivial equations that satisfy

$$a_j^* = a_k^*, b_j^* = b_k^*, c_j = c_k, (i \neq j). \tag{7}$$

Let $\mathbf{x}_1$ be the solution of $a_i^* = 1, b_i^* = 0, c_i = 0 (i = 1, 2, \ldots)$, $\mathbf{x}_2$ be the solution of $a_i^* = 0, b_i^* = 1, c_i = 0$, and $\mathbf{x}_3$ be the solution of $a_i^* = 0, b_i^* = 0, c_i = 1$. Then, the above trivial solutions form a linear space spanned by the bases of $\mathbf{x}_1$,$\mathbf{x}_2$,$\mathbf{x}_3$, which we represent as T.

We describe a numerical solution of the simultaneous equations assuming the observed coordinates (u_i, v_i) on the image plane include errors. Since the equation (6) is over-constrained, the equation generally cannot be fulfilled completely. First, we consider the n-dimensional linear space S_n spanned by the n eigenvectors of $\mathbf{L}^\top\mathbf{L}$ associated with the n minimum eigenvalues. Then, S_n becomes the solution space of $\mathbf{x}$ such that $\max_{\mathbf{x}\in S_n} |\mathbf{L}\mathbf{x}|/|\mathbf{x}|$ is the minimum with respect to all possible n-dimensional linear spaces.

Even if coordinates of u_i, v_i are perturbed by additive errors, $\mathbf{x}_1$,$\mathbf{x}_2$,$\mathbf{x}_3$ remain trivial solutions that completely satisfies equations(5) within the precision of floating point calculations. Thus, normally, the 3D space S_3 becomes equivalent with the space of trivial solutions T. For non-trivial solution, we can define a unit solution $\mathbf{x}_s = \mathrm{argmin}_{\mathbf{x}\in T^\perp}(|\mathbf{L}\mathbf{x}|/|\mathbf{x}|)^2$, where $T^\perp$ is the orthogonal complement space of T. $\mathbf{x}_s$ is the solution that minimizes $|\mathbf{L}\mathbf{x}|/|\mathbf{x}|$ and is orthogonal to $\mathbf{x}_1$,$\mathbf{x}_2$ and $\mathbf{x}_3$. Since T and S_3 are normally equal, $\mathbf{x}_s$ can be calculated as the eigenvector of $\mathbf{L}^\top\mathbf{L}$ associated with the 4-th minimum eigenvalue.

Thus, the general form of the non-trivial solutions are represented as

$$\mathbf{x} = f_1\mathbf{x}_1 + f_2\mathbf{x}_2 + f_3\mathbf{x}_3 + f_4\mathbf{x}_s = \mathbf{M}\mathbf{f}, \tag{8}$$

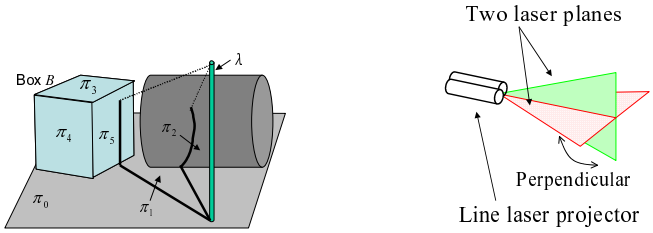
where f_1, f_2, f_3, f_4 are free variables, $\mathbf{f}$ is a vector of $(f_1\ f_2\ f_3\ f_4)^\top$, and $\mathbf{M}$ is a matrix of $(\mathbf{x}_1\ \mathbf{x}_2\ \mathbf{x}_3\ \mathbf{x}_s)$. The four DOFs of the general solution basically correspond to the DOFs of generalized projective bas-relief (GPBR) transformations described in the work of Kriegman *et al.* [8].

As far as we know, there are no previous studies that reconstruct 3D scenes by using the linear equations from the 3-DOF implicit and explicit planes. Advantages of this formulation are that the solution can be obtained stably, and the wide range of geometrical constraints can be used as metric constraints.

3.2 Euclidean reconstruction using metric constraints

The solution obtained in the previous section has four DOFs from $\mathbf{f}$. In addition, if camera parameters are unknown, additional DOFs should be resolved to achieve metric reconstruction. To achieve this, constraints other than coplanarities should be used.

For many scenes, we can find geometrical constraints among explicit and implicit planes. Examples of such information are explained here.



(a) (b)

Figure 2: Metric constraints of coplanarities in a scene:(a) Rectangular box with shadow of straight bar. $\pi_0 \perp \pi_1$ and $\pi_0 \perp \pi_2$ if $\lambda \perp \pi_0$. $\pi_3 \perp \pi_4$, $\pi_4 \perp \pi_5$, $\pi_3 \perp \pi_5$, and $\pi_3 \parallel \pi_0$ if box B is rectangular and on π_0. (b) A laser projector with two line lasers.

1. In figure 2(a), the ground is plane π_0, and linear object λ is standing vertically on the ground. If the planes corresponding shadows of λ are π_1 and π_2, $\pi_0 \perp \pi_1$,$\pi_0 \perp \pi_2$ can be derived from $\lambda \perp \pi_0$.

2. In the same figure, the sides of box$B$ are π_3,π_4, and π_5. If box$B$ is rectangular, π_3,π_4, and π_5 are orthogonal with each other. If box$B$ is on the ground, π_3 is parallel to π_0.

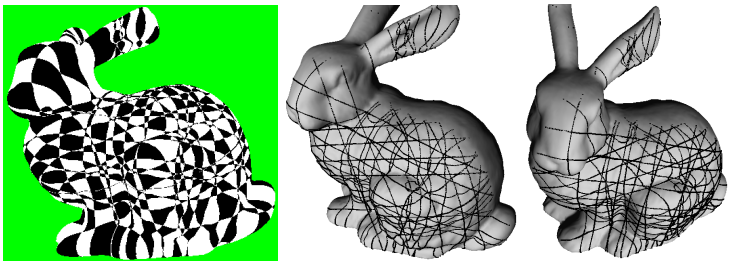
3. Figure 2(b) shows a line projector with two line lasers that are aligned by the right angle. By scanning the scene with this type of projector, orthogonalities between the implicit planes are automatically obtained.

Normally, metric constraints can be represented as nonlinear equations using the free variable vector $\mathbf{f}$ and the unknown intrinsic parameters. To solve these nonlinear equations we use nonlinear optimization. The advantage of nonlinear optimization is that because of the freedom in the definition of the objective function, we can easily deal with many kinds of metric constraints.

To implement a stable nonlinear optimization, we propose a two step optimization. The first step involves optimizing the objective function with respect to the free variable vector $\mathbf{f}$ by using constant intrinsic parameters. The unknown intrinsic parameters are fixed to appropriate initial values in this step. The second step involves optimizing the objective function with respect to both $\mathbf{f}$ and the unknown intrinsic parameters. In many cases, the given information only allows us to reconstruct the scene up to scale. In this case, we fix one of the elements of $\mathbf{f}$, and the optimization is conducted for the rest of the variables.

The determination of the initial value of $\mathbf{f}$ may be a problem. In the experiments described in this study, the initial vector $\mathbf{f}_I$ is calculated from the initial plane parameter $\mathbf{x}_I$ by $\mathbf{f}_I = \mathbf{M}^\top \mathbf{x}_I$. Since $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ and $\mathbf{x}_s$ (column vectors of $\mathbf{M}$) are unit and orthogonal with each other, $\mathbf{M}\mathbf{f}_I = \mathbf{M}\mathbf{M}^\top \mathbf{x}_I$ can be considered as the projection of the $\mathbf{x}_I$ (the vector of initial plane parameters) onto the solution space of the projective reconstruction (8) such that the Euclidean distance between $\mathbf{x}_I$ and $\mathbf{M}\mathbf{f}_I$ is minimum. Using this process, we can obtain a set of plane parameters which fulfills the coplanarity conditions for an arbitrary set of plane parameters.

For example, suppose that the orthogonality between the planes π_s and π_t is assumed. We denote the unit normal vector of plane π_s as a vector function $\mathbf{n}_s(\mathbf{f}, \alpha) =$



(a) (b) (c)

Figure 3: Reconstruction of a CG-synthesized scene: (a) the input image, and (b)(c) the reconstructed scene (the curves) with ground truth (the shaded surface).

$N((a_s(\mathbf{f},\alpha)\ b_s(\mathbf{f},\alpha)\ c_s(\mathbf{f},\alpha))^\top)$ whose parameters are $\mathbf{f}$ and the camera parameter α, where $N()$ means an operation of normalization. Then, the orthogonality between π_s and π_t can be expressed as

$$\{(\mathbf{n}_s(\mathbf{f},\alpha)\}^\top\{\mathbf{n}_t(\mathbf{f},\alpha)\} = 0. \tag{9}$$

Another example of metric constraints is parallelism. Suppose that the planes π_s and π_t are parallel. The parallelism can be expressed as

$$\{(\mathbf{n}_s(\mathbf{f},\alpha)\} \times \{\mathbf{n}_t(\mathbf{f},\alpha)\} = 0. \tag{10}$$

Other than the above objective functions, we can use any functions that are described by the parameters of the points and planes and become minimum for the correct Euclidean reconstruction.

3.3 Dense reconstruction from video

After Euclidean reconstruction of sparse points, a dense 3D shape can be reconstructed by using all the captured frames. The actual process is as follows. First, we detect the intersections between a reflected curve of an unknown implicit laser plane and the curves of already reconstructed laser planes. Since the 3D positions of such intersections are known, we can estimate the parameters of the unknown plane by fitting it to the intersection 3D points using principal components analysis (PCA). We iterate the process for all frames and finally a dense 3D shape can be reconstructed.

4 Experiments

4.1 CG synthesized scene scanned by line lasers

We performed experiments on the reconstruction of 3D scenes with curved surfaces based on the implicit coplanarities. In the experiments, the nonlinear equations obtained from the metric constraints are solved using optimizations based on the Levenberg-Marquardt method.

For the first experiment, we synthesized a test data by CG as shown in 3(a), assuming a laser projector composed of two line lasers, whose laser planes are configured to be

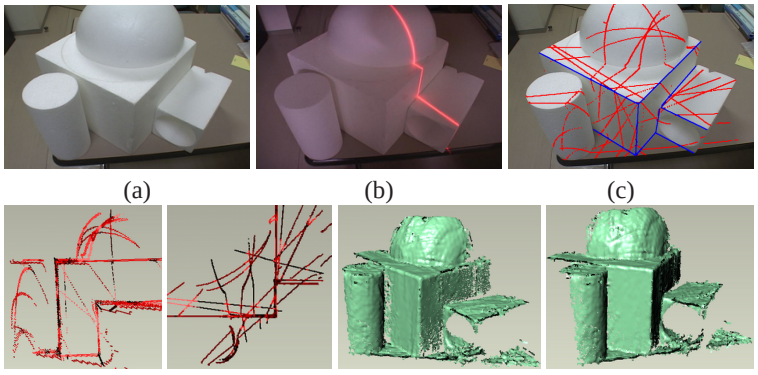


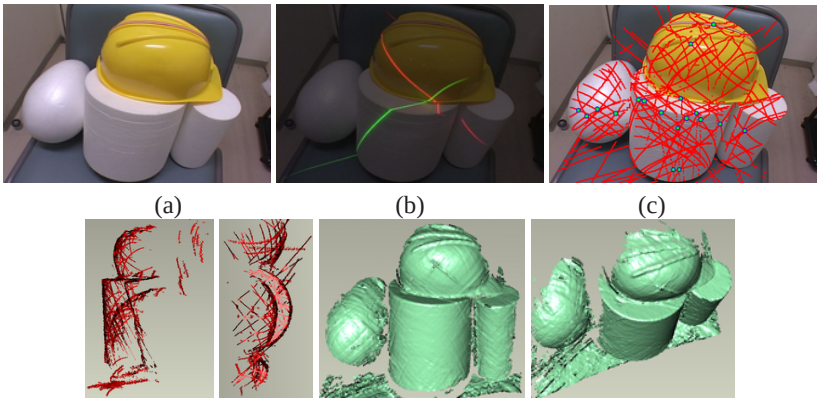
Figure 4: Reconstruction of the real scene from implicit and explicit coplanarities: (a) the target scene, (b) images used to extract the reflections of the line lasers, (c) extracted reflections(red curves) and explicit coplanarities(blue lines), (d)(e) reconstructed scene of line-lasers, and (f)(g) result of dense reconstruction .

perpendicular as shown in figure 2(b). By using the orthogonalities between the laser planes, the scene can be reconstructed without any metric constraints from the scene itself. For this scene, the cross sections of the laser planes and the model were calculated for various positions of the laser projector. The borders of the black and white patterns on the scene represent the cross sections. The images are taken 20 times, and 40 laser planes exist in the scene. The metric constraints are 20 orthogonalities between the planes. The Euclidean reconstruction was performed assuming $\alpha_u = \alpha_v, u_c = v_c = 0$. Since the scaling factor cannot be solved, we represented the solution using the average distance from the camera to the points of the model as the unit length. Using this scale, the bounding box of the ground truth points was $-0.29 \leq x \leq 0.25$, $-0.23 \leq y \leq 0.31$, $-1.34 \leq z \leq -0.93$. Figure 3(b),(c) show the solution (the curves) and the shaded ground truth model. α_u was estimated to be 7.467×10^2, whereas its true value is 7.464×10^2. The RMS of the error was 4.822×10^{-5}; therefore, the reconstruction was very accurate.

4.2 Real scene scanned by line lasers

To conduct experiments for a real object, we use a system consisting of a line laser projector and a video camera. A scanning process is performed by capturing a sequence of images with a fixed camera and moving the line laser back and forth manually. The reflections on the scene are observed as curves, and multiple curves are obtained from the image sequence. Then, we select a few images and detect the cross sections of the reflection curves. By using the points, we can reconstruct projective 3D shapes.

In the first experiment, we used a single line laser. We selected 20 images from a captured image sequence and reconstructed the 3D shape. From the scene, orthogonalities of the faces of the boxes are used as the metric constraints. Figures 4(a)–(e) show the inputs and results. We can clearly observe that the orthogonalities of the rectangular box and the parallelisms of the edges are successfully reconstructed. Then, we conducted a dense 3D reconstruction by using all the captured frames. Figures 4(f) and (g) show the



(a) (b) (c)

(d) (e) (f) (g)

Figure 5: Reconstruction of the real scene from implicit coplanarities: (a) the target scene, (b) images used to extract the reflections of the line lasers, (c) extracted reflections (red curves) and cross sections where metric constraints are imposed(green points), (d)(e) reconstructed line-lasers, and (f)(g) result of dense reconstruction .

recovered dense 3D points. We can confirm that a dense reconstruction with an arbitrary shaped object was achieved.

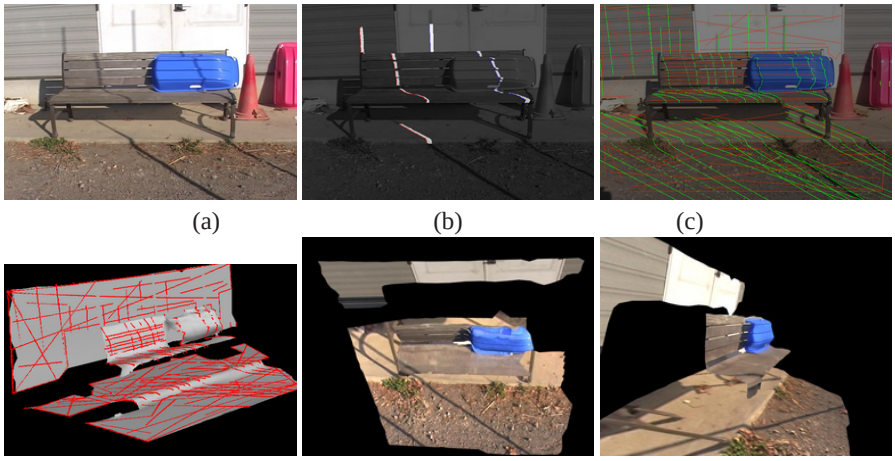
Next, we built a special laser projecting device consisting of two line lasers that were aligned precisely at 90° as shown in Figure 2(b). In this case, no metric constraints were required from the scene. We selected 23 images and reconstructed the 3D shape. We also conducted a dense reconstruction. Figures 5(a)–(g) show all the inputs and results. We can see that an arbitrary shape is successfully reconstructed.

4.3 Real scene reconstruction from shadows of static objects

We conducted a shape reconstruction from images acquired by an outdoor fixed uncalibrated camera. Images from the camera were captured periodically and a shape and the focal length of the camera was reconstructed by the proposed technique from shadows in the scene. Since the scene also contained many shadows generated by non-straight edges, the automatic extraction of shadows based on background subtraction technique was difficult, and thus these noises were eliminated by human interactions. The figure 6 (a) shows the input frame, (b) shows the detected coplanar shadow curves, (c) shows all the coplanar curves and their intersections, and (d) to (f) show the reconstruction result. The proposed technique could correctly reconstruct the scene by using images from a fixed camera.

4.4 Real scene reconstruction from active scan by cast shadows

Next, we conducted an indoor experiment on an actual scene by using a point light source. A video camera was directed toward a target scene of an object of a ceramic jug shaped like a cock and multiple boxes. The target scene was captured to obtain a series of images



(a) (b) (c)

(d) (e) (f)

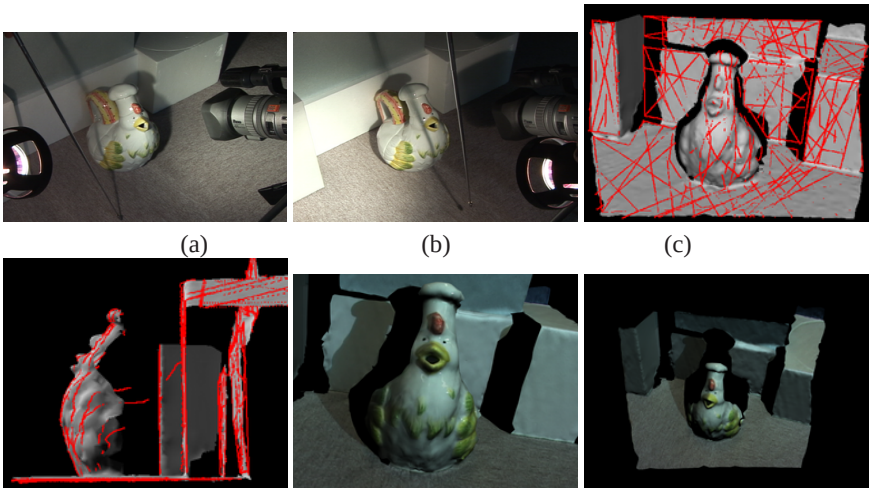
Figure 6: Reconstruction of outdoor scene: (a) input image, (b) a frame of the 3D segmentation result, (c) implicit (green) and explicit (red) coplanar curves, (d) reconstructed result of coplanar curves(red) and dense 3D points(shaded), and (e)(f) the textured reconstructed scene.

while the light source and the bar for shadowing were being moved freely. From the series of images, several images were selected and curves created by the shadow were detected from the images. By using detected coplanar shadow curves, we performed the 3D reconstruction up to 4 DOFs. For the metric reconstruction, orthogonalities of faces of the boxes were used.

Figures 7 (a)-(f) show the capturing scenes and the reconstruction result. In this case, since there were only small noises extracted because of indoor environment, shadow detection based on background subtraction technique worked well and no human interaction was required. The side orthogonalities of the rectangular box and the coplanarities of points on each plane are well reproduced. Unlike 3D photography, the proposed technique realizes reconstruction even if both the light source and the bar are moved freely.

5 Conclusion

In this paper, we propose a novel 3D reconstruction method that utilizes both the coplanarities of points lit by line lasers or those on the boundaries of shadows of straight edges. For obtaining a solution, we first obtain a projective reconstruction by solving the linear equations that are derived from the coplanarity constraints. Then, to upgrade the projective solution to the Euclidean space, we solve the nonlinear equations formulated from the metric constraints, using a nonlinear optimization method. We can use geometrical constraints such as orthogonalities and parallelisms among both the real surfaces and the laser (shadow) planes. By implementing the technique and conducting an experiment using simulated and real images, correct and dense shape reconstruction could be achieved.



(a) (b) (c)

(d) (e) (f)

Figure 7: Reconstruction of an indoor real scene: (a)(b) the capturing scenes, (c)(d) the reconstructed coplanar shadow curves (red) with dense reconstructed model(shaded), and (e)(f) the textured reconstructed model.

References

[1] Didier Bondyfalat, Bernard Mourrain, and Theodore Papadopoulo. An application of automatic theorem proving in computer vision. In *Automated Deduction in Geometry*, pages 207–231, 1998. 1, 2

[2] J. Y. Bouguet and P. Perona. 3D photography on your desk. In *ICCV*, pages 129–149, 1998. 2

[3] Chang Woo Chu, Sungjoo Hwang, and Soon Ki Jung. Calibration-free approach to 3D reconstruction using light stripe projections on a cube frame. In *Third Int. Conf. on 3DIM*, pages 13–19, 2001. 2

[4] R. B. Fisher, A. P. Ashbrook, C. Robertson, and N. Werghi. A low-cost range finder using a visually located, structured light source. In *Second Int. Conf. on 3DIM*, pages 24–33, 1999. 2

[5] Ryo Furukawa and Hiroshi Kawasaki. Interactive shape acquisition using marker attached laser projector. In *Int. Conf. on 3DIM2003*, pages 491–498, 2003. 2

[6] E. Grossmann, D. Ortin, and J. Santos-Victor. Single and multi-view reconstruction of structured scenes. In *ACCV*, pages 93–104, 2002. 1, 2

[7] Etienne Grossmann, D. Ortin, and José Santos-Victor. Algebraic aspects of reconstruction of 3d scenes from one or more views. In *BMVC*, 2001. 1, 2

[8] David J. Kriegman and Peter N. Belhumeur. What shadows reveal about object structure. *Journal of the Optical Society of America*, 18(8):1804–1813, 2001. 4

[9] D. Liebowitz and A. Zisserman. Metric rectification for perspective images of planes. In *CVPR*, page 482, 1998. 1, 2

[10] Peter Sturm and Steve Maybank. A method for interactive 3d reconstruction of piecewise planar objects from single images. In *BMVC*, pages 265–274, Sep 1999. 1, 2

[11] Kokichi Sugihara. An algebraic approach to shape-from-image problems. *Artificial Intelligence*, 23:59–95, 1984. 1, 2

Tracking Using Online Feature Selection and a Local Generative Model

Thomas Woodley * Bjorn Stenger † Roberto Cipolla *

* Dept. of Engineering
University of Cambridge
{tew32|cipolla}@eng.cam.ac.uk

† Computer Vision Group
Toshiba Research Europe
bjorn@cantab.net

Abstract

This paper proposes an algorithm for online feature selection which improves robustness to occlusions by referring to a localized generative appearance model. Discriminative classifiers based on feature extraction have classically either prepared a fixed prior model by training offline, or continually adapted their classification parameters to any apparent appearance changes. By combining the attractive qualities of each approach, our framework can cope with appearance changes of a target object and will maintain proximity to a static appearance model. Our main contribution is the use of a generative model to guide the online feature selection to regions of an image which maintain a valid appearance. The generative model exhibits the properties of non-negativity, localization and orthogonality. We demonstrate the system in a tracking framework to show improved tracking performance through occlusions.

1 Introduction

A major challenge in visual tracking is handling the appearance variation of the target object. This can be caused by a number of factors including pose variation, shape deformation, lighting changes, as well as occlusions. In this paper we follow a discriminative approach to tracking where a classifier is used to distinguish the object from the background. The classifier uses a set of discriminative local features which is updated at each time step using on-line boosting [4]: using the previous object location as positive example and surrounding regions as negative examples the classifier updates its feature pool and corresponding weights. This flexibility is advantageous for tracking through large appearance changes but leads to the known *template update problem* [14]. The key question is how much adaptability to allow the tracker. In other words: How can one decide whether an appearance change is simply due to pose or lighting variation or due to occlusion of the object?

One strategy is to use a generative object model and determine whether the current target estimate is still valid given this model. One such model representation is an eigenspace model, where the image is modelled by a linear combination of orthogonal basis functions. The model can be used to statistically evaluate the presence of the object. However,

as it is a global object representation it does not provide a straightforward way to estimate local occlusions.

In this paper we introduce a method that uses a *local* generative model to constrain the selection of local features in the classifier in the case of outliers caused by local occlusion. Our generative model is computed from the first few frames in the sequence by local non-negative matrix factorization (LNMF) where the basis functions are orthogonal and sparse and spatially constrained. The key idea in this paper is to identify occluded regions by projecting the current image onto the basis functions. Such outliers are determined by comparing with the distribution of individual coefficients. This information is integrated with online feature selection as follows: If a region is labelled as occluded, the local features in this region are discarded and new features in the non-occluded regions added. Alternatively, the features in the occluded regions can be de-activated for the duration of the occlusion. This way the adaptation to outlier regions is avoided while being able to keep valid classifiers in memory that have large feature support in the target region.

The rest of the paper is laid out as follows: Section 2 discusses related prior work. In section 3 we introduce a new algorithm for combining discriminative and local generative models for improved online feature selection. We report on experiments carried out to verify the approach in section 4 while section 5 concludes with a discussion of the contributions.

2 Prior Work

Adaptive object tracking is a core technique in many applications and has therefore been widely explored. We provide a brief summary of the work most relevant in the context of this paper.

Tracking using classifiers Avidan introduced the idea of using a binary classifier to track an object [1]. A Support Vector Machine (SVM) classifier is trained off-line to discriminate between object and background. Tracking is carried out by estimating transformation parameters that maximise the SVM score. This idea was extended by Williams et al. [19] who provided a probabilistic formulation allowing to propagate observation distributions over time. A mapping from image space to transformation parameter space is learned from seed images. These methods are quite robust, however they rely on the appearance variation being fully encoded within the classifier, which is not updated once the tracker is running. Collins et al. [3] proposed to update a classifier by selecting a set of discriminative features in each frame. It is assumed that the object was correctly located and that the surrounding area belongs to the background. The idea of updating a classifier based on AdaBoost using discriminative feature selection was introduced by Grabner and Bischof [4]. In [5] they show tracking over large appearance variation. However, neither of these methods [3, 5] maintains an explicit object model to prevent drift or adaptation to outliers.

Tracking using subspace models Subspace models have been used to model object appearance. The idea is that the images of a particular object lie on a lower dimensional manifold. The eigenspace tracking approach was introduced by Black et al. in [2]. Since then the idea has been extended to handling appearance changes. Several methods have

been proposed to learn an eigenspace representation and incrementally update it over time [6, 13, 17]. Lee and Kriegman adapt a generic appearance model to a specific one given a number of images [10]. Appearance is modelled by a union of linear manifolds and the model is updated by incrementally updating their eigenbases. None of these papers explicitly addresses outlier handling, although robust error norms may be able to handle some cases of partial occlusion.

Outlier handling Jepson et al. [8] use an adaptive appearance model where each pixel intensity is modelled by a three-component mixture. One of the components ('lost') is used to handle outliers caused for example by occlusion. Williams et al. [18] compute an outlier mask using a Markov Random Field with Ising prior in order to increase the tolerance of a foreground/background classifier to occlusions. The estimates are quite accurate, however this method of outlier handling adds significant computation overhead. Outlier handling is also being investigated in contexts other than tracking. Leonardis and Bischof [11] modified the eigenspace approach for recognition to handle partial occlusions. Instead of computing the PCA coefficients by projecting the complete image, the coefficients are found using only a subset of image points. Several subsets of points are hypothesized and tested using the backprojection error. Oh et al. [15] subdivide a face image into regions and construct separate eigenmodels for each. The occluded regions are found by separately computing the distance to the nearest input training sample. Recognition is performed on the occlusion free regions.

3 Discriminative Feature Selection Using A Local Generative Model

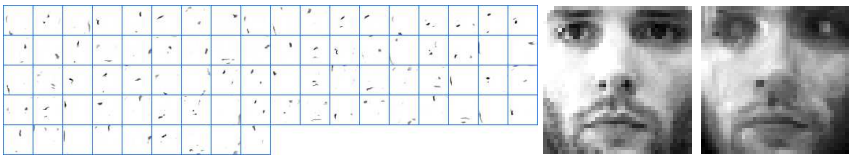
In this section we first review the principles of online boosting and local non-negative matrix factorization. We then show how to combine these two techniques to improve an adaptive tracking algorithm in the case of local occlusion.

3.1 Online tracking with AdaBoost

Online boosting for tracking has recently been introduced by Grabner et al. [4, 5]. The principle is to locate the object by maximizing the score of a boosted classifier at every time step. Following the localization step the classifier itself is updated with online boosting. The target estimate is used as a new positive example and surrounding regions as negative examples. The classifier, called a strong classifier, is built by a linear combination of a number of weak classifiers, where each weak classifier corresponds to evaluating a single feature. The features are chosen from a global feature pool by the online feature selection process based on their classification performance so far. With each update the algorithm uses the new training sample to compute features and classification weights with which to compute the updated strong classifier. For a further details, see [4, 5].

3.2 Local non-negative matrix factorization

Non-negative matrix factorization (NMF) is a method for finding a lower dimensional representation of data. Given a non-negative data matrix $\mathbf{X}$, NMF finds an approximate



(a) (b) (c)

Figure 1: **Local non-negative factorization on face images.** This figure shows **(a)** basis images found by LNMF. Note that the positive entries are sparse and localized. The basis images are used to approximate an input image **(b)**, the result is shown in **(c)**.

factorization $\mathbf{X} = \mathbf{BH}$ into non-negative matrices $\mathbf{B}$ and $\mathbf{H}$, i.e. their elements must be equal to or greater than zero [16]. Lee and Seung compared NMF to principal component analysis (PCA) and vector quantization (VQ), and showed that these can be written as factorizations with different constraints [9]. NMF is able to learn a parts-based representation while PCA and VQ both learn holistic representations which in the case of PCA can also contain negative entries. Several extensions of the original NMF algorithm have been proposed, in particular versions that impose a sparseness constraint on the matrix $\mathbf{H}$ [7, 12]. In this section we follow the exposition of Li et al. [12], who suggested an algorithm for local non-negative matrix factorization (LNMF) of images: Writing a set of N_T images into an $n \times N_T$ matrix $\mathbf{X}$, so that each column consists of the n pixel values, LNMF factorizes this matrix into an $n \times m$ matrix $\mathbf{B}$ containing a set of $m < n$ basis images, and an $m \times N_T$ coefficient matrix $\mathbf{H}$, such that $\mathbf{X} \approx \mathbf{BH}$. Additionally, LNMF imposes the following three constraints: (i) Maximum sparsity in $\mathbf{H}$, (ii) maximum expressiveness of $\mathbf{B}$, and (iii) maximum orthogonality of $\mathbf{B}$. A locally optimal solution is found by iteratively updating $\mathbf{B}$ and $\mathbf{H}$. See [12] for details. Once the subspace images are found, an image represented as an n-vector $\mathbf{x}$, is projected into the space by $\mathbf{h} = \mathbf{B}^+\mathbf{x}$, where $\mathbf{B}^+$ denotes the pseudo-inverse of $\mathbf{B}$. Figure 1 shows basis images found by LNMF (the columns of $\mathbf{B}$) as well as an example image where these basis images are used for approximating an input image.

3.3 Feature selection using a local generative model

The orthogonality of the basis images found with LNMF implies that the value of a pixel x_i in the decomposition of an image $\mathbf{x}$ is determined solely by the positive value in one of the basis images at the corresponding location i, and the corresponding weight, that is

$$x_i \approx \sum_j h_j b_{ij} \approx h_k b_{ik} \quad , k \in \{0, \ldots, m\}. \tag{1}$$

From the training images we obtain distributions $p^{fg}(h_j)$ for the weights of each dimension j of the subspace. We model the weight distribution in each dimension j with a Gaussian mixture which is used to determine the foreground likelihood. Thus we have a means of determining how well a new image locally matches the appearance model and we can create a foreground likelihood map for a new input image. If the likelihood is below a given threshold value, the corresponding regions are treated as outliers. In practice, LNMF factorization can result in more than one disconnected component in a basis

Algorithm 1 Online feature selection using a local generative model

Input: New image as n-vector $\mathbf{x}$, projection matrix $\mathbf{B}^+$ (computed off-line), coefficient inlier distributions $p^{fg}(h_j)$ (computed off-line), threshold values θ_{fg}, θ_2.

1. Compute LNMF coefficients $\mathbf{h} = \mathbf{B}^+\mathbf{x}$.
2. Initialise outlier map as n-vector $\mathbf{c} = 0$.
3. for each coefficient h_j
   if $p^{fg}(h_j) < \theta_{fg}$
   set $c_i = 1$ for $\{i \mid b_{ij} > 0\}$
4. Remove small connected components in $\mathbf{c}$.
5. for each feature f_k in classifier feature pool
   if $\sum_{i \in support(f_k)} c_i > \theta_2$
   replace f_k with new feature in non-occluded region
6. Update classifier using the online AdaBoost algorithm [5].

image, with some components being very small (see Fig. 1a). Under the assumption that the true region of outliers in an image is approximated by the union of basis image components, we threshold the binary image on component size to remove small components. This information is used to guide the online feature selection: Local features whose support region overlaps more than a threshold value with outlier regions are excluded from the classifier. Thus the algorithm only considers features in regions consistent with the generative model, and the discriminative classifier avoids locking on to occluded or background regions. Additionally, we cache classifiers in cases that are not occluded and use them to regain lock after the target has been occluded significantly. Algorithm 1 details the adapted online learning algorithm.

4 Experimental results

This section presents experimental results to validate our algorithm. In all experiments the appearance model was created by resampling the training images to 40x40 pixels, and factorizing with a subspace dimension of 81 in order to achieve sufficient localization. The LNMF model is learned from training sequences containing approximately 700 frames. Once the subspace basis is computed tracking is executed in real-time (around 50 ms per update). The strong classifier uses 50 selectors and the feature pool contains 250 weak classifiers. These are the same settings as in [5], however we only use local rectangle features ('Haar-like features') to demonstrate the improved feature selection. All experiments were carried out on a standard 3.4 GHz PC with 2GB RAM.

4.1 Detecting occlusions

As a proof of concept we show the distribution of coefficients h_j for one particular basis image over a sequence with significant occlusion. Figure 2 shows the results. One can observe a shift in the distribution during the time of occlusion. This shows that occluded regions can be detected by a change in the weight values of the basis images with positive values in those regions.

Figure 2: **Shift of coefficient distribution under occlusion.** *The histogram (left) shows the distribution of weights for one of the LNMF basis functions. There is a shift during the occluded case. The images show from left to right: basis image (white=zero, dark=positive), an unoccluded example, an example with occlusion intersecting the positive support region of the basis image.*

Figure 3: **Online feature selection under occlusion.** *The model helps to avoid erroneous feature selection in the occluded regions. Row 1 shows sample frames. Row 2 shows regions detected as occluded (red) and support regions of features used in the classifier (highlighted).*

We perform the adapted online feature selection algorithm (see Alg. 1) on a number of test sequences. Figure 3 shows sample frames in the top row and the algorithm output below. The occlusion map is shown in red and the support area of features currently used in the classifier is highlighted in white. We can see there is agreement between the occlusion map and the actual occlusion. Further, the feature selection is restricted to those areas labelled as not occluded. Figure 4 shows an example sequence with a large pose change. In such cases the target image can contain background regions. The outlier detection prevents feature selection in these regions.

4.2 Synthetic occlusion

We take a single test image, and create a test sequence by adding fixed size, randomly positioned black squares to simulate occlusion. Figure 5 shows the accuracy of the occlu-

Figure 4: **Feature selection under pose change.** *Large pose changes can lead to large regions of background being inside the target window. By labelling these as outliers, we avoid adaptation to the background. Row 1 shows input frames. Row 2 shows regions detected as occluded (red) and support regions of features (highlighted).*

sion detection with varying occlusion size from 5×5 to 155×155 pixels within a window of size 160×160 pixels. We measure the true positive rate and false positive rate in terms of pixel classification rate for each frame in a test sequence of random fixed-size occlusions and take the average to plot a point on the curve. Additional points are generated by running test sequences of differing occlusion size. The detection rate increases with the size of the occluded region. However, larger occlusions lead to an increase in false detections. This is due to the fact that there is no perfect alignment between occluded regions and support regions of basis images.

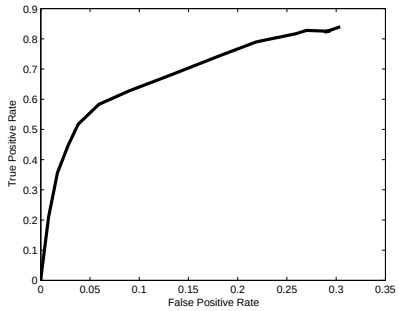


Figure 5: **Accuracy for different sizes of occlusion.** *Occlusion detection accuracy against occlusion size (increasing from 5×5 bottom left to 155×155 top right). Occlusion detection increases with size of occluded region but with the consequence of an increased false detection rate.*

Figure 6: **Real-time tracking with occlusion.** *Row 1 shows tracking results with feature selection using online boosting. The flexibility of the classifier leads to adaptation to the occluding object. Row 2 shows corresponding results with the new algorithm.*

Figure 7: **Real-time tracking during occlusion.** *Tracking continues during partial occlusion. During occlusion new features are allocated in the visible regions allowing tracking to continue. When the occluding object is removed the whole region is again used for feature selection.*

4.3 Tracking results

We run the tracker on test sequences containing heavy occlusion, both without and with the adapted online feature selection. The original tracker adapts to the appearance of the occluding object, and starts to track this. With our adapted feature selection the tracker is able to note the occlusion and stop feature selection in the occluded areas, regaining lock when the target re-appears. Figure 6 shows sample frames from the tracking sequence: row 1 shows the original tracker, row 2 shows the tracker using the results from our adapted online feature selection. Figure 7 shows continued tracking during partial occlusion. During occlusion new features are allocated in the visible regions. When the occluding object is removed the whole region is again used for feature selection.

Figure 8 shows tracking results on a publicly available sequence [8]. Row 1 shows the tracker using online boosting. The sequence was tracked successfully in [5] using a larger variety of features. Note that when using rectangle features only we observe a small shift in the target region estimate. Row 2 shows the result of our proposed algorithm. The occlusion of the face by the hand is detected correctly and tracking continues successfully.

Figure 8: **Real-time tracking with occlusion.** *Tracking results for the public 'Dudek' sequence [8]. Row 1 shows sample tracking results without using our adapted online feature selection. Row 2 shows corresponding results with the new algorithm. Regions detected as occluded are coloured white. Note that in contrast to [5] we use rectangle features only.*

5 Summary and conclusions

In this paper we have shown how a localized generative appearance model can be used by an online learning algorithm to focus feature selection on regions in an image which maintain a good proximity to the target object's appearance. We have demonstrated how this approach can improve the robustness of an adaptive tracker to occlusions while maintaining real-time performance. To our knowledge this is the first time that local non-negative matrix factorization has been employed within an object tracking context.

It can be noted that the outlier detection is dependent on the regions of positive value in the basis images, which we have no explicit control over in LNMF. There is also a trade-off between adaptiveness and outlier detection: A training sequence which shows little variation will result in good outlier region detection due to the smaller variance of the weight values. However, adaptiveness will be limited due to the tighter bounds on the appearance model. Conversely, a training sequence with large appearance variation will allow a lot more adaptiveness, but be less capable of detecting outlier regions. Future work will focus on a more thorough evaluation of the tracker, and increasing the flexibility of the appearance model.

References

[1] S. Avidan. Support vector tracking. *Trans. Pattern Analysis and Machine Intelligence*, 26(8):1064–1072, 2004.

[2] M. J. Black and A. D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *Int. Journal of Computer Vision*, 26(1):63–84, 1998.

[3] R. T. Collins, Y. Liu, and M. Leordeanu. Online selection of discriminative tracking features. *Trans. Pattern Analysis and Machine Intelligence*, 27(10):1631–1643, 2005.

[4] H. Grabner and H. Bischof. Online boosting and vision. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 1, pages 260–267, 2006.

[5] H. Grabner, M. Grabner, and H. Bischof. Realtime tracking via online boosting. In *Proc. British Machine Vision Conference*, volume 1, pages 47–56, 2006.

[6] J. Ho, K. Lee, M. Yang, and D. Kriegman. Visual tracking using learned linear subspaces. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 1, pages 782–789, 2004.

[7] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *J. Machine Learning Research*, pages 1457–1469, 2004.

[8] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi. Robust online appearance models for visual tracking. *Trans. Pattern Analysis and Machine Intelligence*, 25(10):1296–1311, 2003.

[9] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

[10] K.-C. Lee and D. Kriegman. Online learning of probabilistic appearance manifolds for video-based recognition and tracking. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 852–859, 2005.

[11] A. Leonardis and H. Bischof. Dealing with occlusions in the eigenspace approach. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 453–458, 1996.

[12] S. Li, X. Hou, and H. Zhang. Learning spatially localized, parts-based representation, 2001.

[13] J. Lim, D. Ross, R. Lin, and M. Yang. Incremental learning for visual tracking. In *Advances in Neural Information Processing Systems*, pages 793–800, 2005.

[14] I. Matthews, T. Ishikawa, and S. Baker. The template update problem. *Trans. Pattern Analysis and Machine Intelligence*, 26:810–815, 2004.

[15] H. J. Oh, K. M. Lee, S. U. Lee, and C.-H. Yim. Occlusion invariant face recognition using selective LNMF basis images. In *Proc. Asian Conf. on Computer Vision*, pages 120–129, 2006.

[16] P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5:111–126, 1994.

[17] D. Ross, J. Lim, and M. Yang. Adaptive probabilistic visual tracking with incremental subspace update. In *Proc. Europ. Conf. on Computer Vision*, volume 2, pages 470–482, 2004.

[18] O. Williams, A. Blake, and R. Cipolla. The variational ising classifier (VIC) algorithm for coherently contaminated data. *Advances in Neural Information Processing Systems*, pages 1497–1504, 2004.

[19] O. Williams, A. Blake, and R. Cipolla. Sparse Bayesian learning for efficient visual tracking. *Trans. Pattern Analysis and Machine Intelligence*, 27:1292–1304, 2005.

An Evaluation of Shape Descriptors for Image Retrieval in Human Pose Estimation

Phil Tresadern
CRHPR
University of Salford
Salford M6 6PU, UK
p.tresadern@salford.ac.uk

Ian Reid
Active Vision Lab
University of Oxford
Oxford OX1 3PJ, UK
ian@robots.ox.ac.uk

Abstract

This paper presents an empirical comparison of several shape representations in order to search a database of training examples (silhouettes) for the task of human pose estimation. In particular, we compare the Discrete Cosine Transform (DCT), Lipschitz embeddings and the Histogram of Shape Contexts that has previously demonstrated some success in this task. Our results suggest that a simple linear transformation of the image (such as the DCT) is as effective as the more complex, non-linear methods.

1 Introduction

Due to the rapid increase in affordable secondary storage over the last few years, it is becoming increasingly important to develop systems that retrieve data based on *content* rather than annotating the data by hand. This has led to the growth of interest in shape matching and retrieval algorithms with applications including searching the Web (*e.g.* Google Images) and more specific fields such as trademark enforcement. Since it is typically infeasible to use the raw, high-dimensional image to describe the data, D features are computed that retain the most informative data in the image. This dimensionality reduction provides three major benefits:

- **Lower storage requirements:** each image is reduced to a compact feature vector.
- **Increased efficiency:** the training data can be processed more rapidly.
- **Reduced sensitivity to noise:** features capture the most informative shape characteristics whilst ignoring irrelevant details.

In this work, we compare three shape representations that reduce the dimensionality of training images for the purpose of image retrieval in human pose estimation. In particular, we compare the recently proposed Histogram of Shape Contexts [1] with two simpler descriptors, namely the Discrete Cosine Transform (DCT) and Lipschitz embeddings. Although the success of the Histogram of Shape Contexts for recovering human pose was demonstrated within a sparse regression framework [1], resulting in its adoption in other studies (*e.g.* [10]), to date no empirical evidence has been presented to support claims that this is due to the efficacy of the descriptor rather than the regressor. This work presents the first quantitative comparison to investigate this claim by comparing representations *under controlled conditions where meaningful comparisons can be made*.

1.1 Related Work

The range of shape descriptors available for applications such as human pose estimation from binary silhouettes is very large. However, we can argue that many representations are inappropriate for this task. Descriptors based on the topology of the occluding contour [7] change dramatically with small changes in underlying pose (*e.g.* as the subject places their hands on their hips such that 'holes' are created that modify the topology). Representations based on curvature [15] typically require a continuous (or sufficiently high resolution) contour that is rarely available in this application. Similar arguments rule out Fourier decompositions [16] and shock graphs/median axis representations [9].

Of the remaining candidates, *global* representations use every pixel to compute every feature such that a localized corruption of the input image (*e.g.* due to occlusion or shadow) induces an error in every feature. Such representations include embeddings [5], moments [8, 12, 14] and Principal Component Analysis (PCA). In contrast, *local* representations use only a subset of the image to compute each feature such that only certain features are affected by a localized error in the input image. Such representations include the recently proposed Histogram of Shape Contexts (HoSC) that has successfully been employed in human pose estimation [1]. It is this property of locality that is claimed to make such representations superior.

1.2 Paper structure

We begin in Section 2 by describing the selected shape descriptors, including a discussion of how appropriate parameters were selected for each. Section 3 describes the experimental data and how the descriptors were compared. Results are presented in Section 4.

2 Shape representation

2.1 Discrete Cosine Transform (DCT)

We begin with a form of the Discrete Cosine Transform of the $P \times Q$ image, $I(x,y)$, whereby each feature (DCT coefficient), M_{mn}, is defined by:

$$M_{mn} = \sum_x \sum_y f_m(x) I(x,y) f_n(y) \tag{1}$$

and we define

$$f_m(x) = \sqrt{\frac{1+\min(m,1)}{P}} \cos\left\{ \frac{m\pi}{P} \cdot \left(x + \frac{1}{2} \right) \right\} \tag{2}$$

where $m = 0 \ldots P-1$ and $x = 0 \ldots P-1$. This transform can be an interpreted as a rotation of the vectorized image such that the Euclidean distance between feature vectors in PQ-dimensional space is equal to the sum of squared error between the original images. Using only a subset of D coefficients therefore approximates the SSE between images. Furthermore, this form of the DCT belongs to the family of *orthogonal moments* since:

$$\int f_i(x) f_j(x) dx = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \tag{3}$$

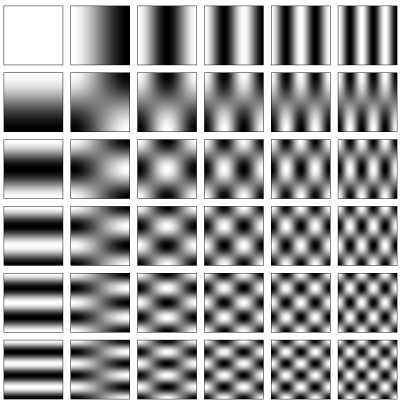


Figure 1: Filter bank equivalents (up to order 5) of DCT moment generating functions, $f_{mn}(x,y) = f_m(x)\,f_n(y)$.

such that correlation is low between coefficients and fewer are required (compared to non-orthogonal moments) to describe the image within a given error bound.

Other transformations were also considered such as Tchebichef [8], Krawtchouk [14], geometric and Hu [6] moments in addition to PCA. Although PCA provides an optimal (in terms of capturing maximum variance) basis set over the set of images, the basis set is data-dependent and impractical to compute for the image sizes involved. Tchebichef moments were found to be qualitatively similar to the DCT, effectively providing a frequency decomposition of the image, although with slightly worse performance in the evaluation task. Krawtchouk moments (another orthogonal moment) also performed slightly worse than the DCT, possibly as a result of limited spatial support of lower order moments.

Geometric moments are seldom employed due to the concentration of 'mass' at the edges of the image (where the least informative data resides) and the lack of an intuitive distance metric between feature vectors (in contrast to orthogonal moments). Similarly, although Hu moments are popular due to their rotational invariance they are based on geometric moments and hence suffer the same shortcomings. Furthermore, only seven Hu moments are typically defined which do not capture sufficient variation in many datasets.

In order to make the comparison fair, we first undertook a number of experiments to assess the impact of various parameters [13]. These experiments suggested that:

- Although performance improved as more DCT coefficients were retained (since the distance between feature vectors more closely approximates the true SSE between images), most useful information was captured by $D \geq 64$ features.
- When ranking the database in order of similarity to the query in feature space, Euclidean distance (the most intuitive metric since it is directly related to the SSE) gave very similar performance to the Mahalanobis and Manhattan (L_1) distances.
- Feature selection heuristics such as maximum order ($\max\{m,n\}$), order ($m+n$) and RMS value all gave similar results whilst variance was a poor indicator of feature information. More complex feature selection is beyond the scope of this work.

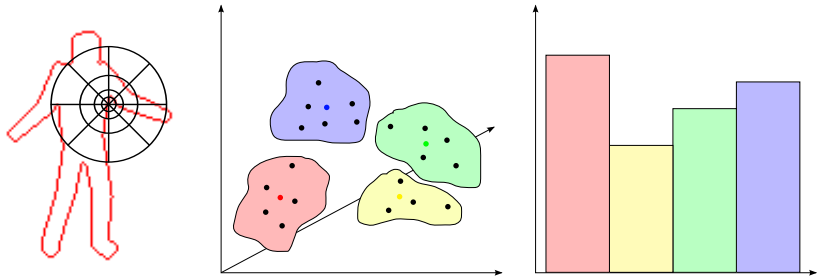


Figure 2: Overview of HoSC descriptor: (a) Each contour point is assigned a high-dimensional 'Shape Context' based on the local distribution of other contour points; (b) Shape Contexts from all database examples are clustered to generate D cluster centres (codebook vectors); (c) A normalized histogram is generated for each example based on the distribution of cluster centres voted into by the Shape Contexts of its contour points.

2.2 Lipschitz embeddings

The second global representation we consider is the Lipschitz embedding [5], whereby an image is represented by the vector of distances from the query image to D 'pivot' exemplars and has recently demonstrated success in hand tracking applications [3]. More specifically, we embed each image by extracting its contour points and computing its (asymmetric) chamfer distance from the ith pivot examplar to give the ith element of the feature vector. Intuitively, images that are close together in image space have similar distances to the pivot examples and therefore have similar feature vectors. However, selecting pivots from the same region of space results in highly correlated (*i.e*. redundant) features that may degrade performance.

Experiments to investigate the effect of various parameters [13] suggested that:

- Most information for this dataset was captured using $D \geq 100$ features (pivot examplars).
- Due to the non-linear nature of the Lipschitz embedding, it is difficult to identify an intuitive distance metric between two feature vectors. However, using the Mahalanobis distance resulted in a noticeable improvement over the Euclidean and Manhattan metrics.
- No significant difference in performance was observed over 100 randomly selected sets of exemplars although a more intelligent approach to feature selection was recently investigated using Boosting [2].

2.3 Histogram of Shape Contexts (HoSC)

Our final selected shape descriptor is the Histogram of Shape Contexts, suggested by Agarwal and Triggs [1], and demonstrated using silhouettes of the human body. In this representation (see Figure 2), every point along the contour of the silhouette is assigned a histogram (known as its Shape Context [4]) representing the distribution of other contour

Figure 3: In this example, both the angel and the demon are composed of identical contour segments such that their histograms become indistinguishable as the spatial extent (*i.e*. the radius) of the shape context vector approaches zero. Note that *exact* tesselation is not required for very different silhouettes to result in very similar feature vectors.

points in a local neighbourhood (defined by the Shape Context 'radius'). Having computed the Shape Context for all contour points on all silhouettes in the database, D Shape Contexts are then selected at random and used as initial centres in a k-means clustering scheme. Following clustering, the updated cluster centres are used as a vector quantization 'codebook' in order to assign each contour point on a given silhouette to a cluster. A histogram over cluster assignments then forms the feature vector for a given silhouette. This histogram should be normalized with respect to the number of contour points to make the descriptor scale-invariant. Furthermore, in order to reduce quantization effects, 'soft' voting allows each contour point to vote into more than one bin.

It is suggested that this descriptor may be superior due to its locality – corrupting a small region of the silhouette should modify only a few features, in contrast to the DCT and Lipschitz embeddings where the whole silhouette contributes to every feature. However, we note that: (i) in most cases the corruption of the silhouette (*e.g*. due to shadows or occlusion) results in an increase or decrease in the number of contour points such that normalizing the histogram then affects *every* bin; (ii) typical distance metrics (*e.g*. Euclidean distance, Bhattacharyya coefficient) do not exploit this locality in any beneficial way; (iii) no explicit distinction is made between the interior and exterior of the silhouette, thus discarding potentially valuable information (see Figure 3).

These concerns provided the motivation behind comparing the Histogram of Shape Contexts to other descriptors in order to quantify any benefit gained from the substantial increase in computational complexity. As with the other descriptors, a basic analysis of the parameters [13] suggested that:

- Again, most information was captured by $D \geq 64$ features (codebook vectors).
- The use of intuitive distance metrics for histograms (*e.g*. Bhattacharyya distance) did not significantly improve performance over other (less correct) metrics such as the Manhattan and Euclidean distance (this has previously been attributed to 'soft' voting [1]).
- Since codebook vectors are typically well distributed after clustering, performance was largely insensitive to their initial random selection as evaluated over 100 trials.

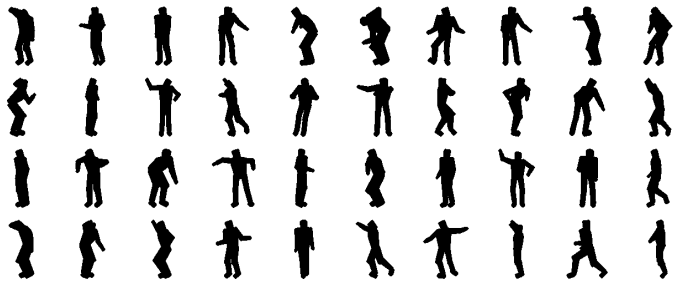


Figure 4: Example silhouettes from the synthetic dataset.

- Performance was stable for any sensible Shape Context 'radius' of at least the mean distance between all pairs of contour points.
- Although we used 12 angular bins (a common value), performance is stable for any value above 8. Performance was largely invariant to the number of radial bins.
- The use of 'soft' voting (as advised in [1]) to avoid quantization effects provided a small benefit when each contour point voted into > 4 bins.

3 Method

In order to evaluate the selected shape descriptors, we used motion capture data (available at the time of printing from `http://mocap.cs.cmu.edu`) to generate N=10000 128×128 binary silhouettes of a human body model (Figure 4). This training set included synthetic silhouettes from several different 'exercise' motions generated from 4 camera locations equally spaced from 0° to 90° in azimuth.

In addition to the training data, an additional 250 silhouettes were generated from synthetic data to test the retrieval performance of the shape descriptors. Furthermore, 40 real test images were obtained by background subtraction of several sequences of a subject undertaking exercise motions similar to those in the training data.

For the purposes of this evaluation, all images were normalized by translating and scaling the silhouette such that it lay within the central 90% of the image. We also assumed that the subject was upright in the image to avoid any need for rotation invariance; any exceptions to this rule (*e.g.* handstands, cartwheels) were explicitly modelled in the dataset. All silhouettes were then reduced to a feature vector of $D = 100$ dimensions using each of the proposed descriptors.

Silhouettes generated from synthetic data were automatically labelled with the image projections of the joint centres since these values were directly available. For silhouettes obtained from real sequences, the image projections of joint centres were labelled manually using the mouse in order to evaluate performance.

Like many other studies, we employ silhouettes since they are readily obtained from image data by background subtraction and are relatively invariant to clothing and lighting. However, they are generally restricted to scenes with a static camera and known background, and useful image data (*e.g.* internal edges) are discarded.

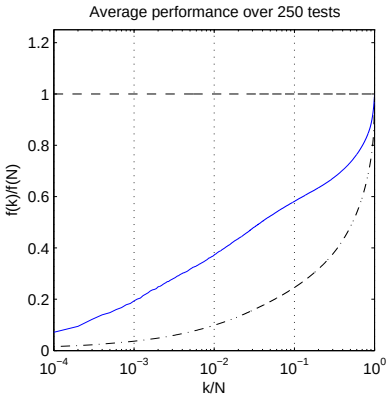


Figure 5: Example graph of k/N against $f(k)/f(N)$. For comparison, the dashed line at unity indicates the average curve produced by random ordering whilst the dash-dot curve indicates the best possible ranking where distance in image space correlates perfectly with distance in pose space.

3.1 Evaluation method

Image retrieval tasks typically require *classification* of the query input such that stored examples of the same class are returned. Recovered exemplars are therefore classed as positive or negative and evaluation tools such as the Receiver Operating Characteristic (ROC) curve and Precision-Recall curve may be used to compare retrieval accuracy between different shape descriptors.

In the context of human pose estimation, however, exemplars cannot be classified into 'positives' and 'negatives' since the underlying pose space is continuous. Therefore, we use the sum of squared errors between corresponding joint centre projections[1] in the image to compute the distance, $d(x_i, x_q)$, in pose space between each training example, x_i, and a query, x_q. Given a query silhouette, we rank the training data in order of similarity to the query as quantified by the chosen shape descriptor, denoting the index of the closest training example by $r(1)$ and the furthest by $r(N)$. We then generate a curve, $f(k)$:

$$f(k) = \frac{\sum_{j=1}^{k} d(x_{r(j)}, x_q)}{k}, \tag{4}$$

indicating the mean distance to the query of the k highest ranking training examples for $k = 1 \ldots N$. For a qualitative performance evaluation, we compare the normalized curve of k/N against $f(k)/f(N)$ in addition to the corresponding curves for the expected performance of a random ranking of the training data (*i.e.* unity) and for the best possible ranking, as shown in Figure 5. Each curve can be interpreted as a measure of correlation between distance in state space and distance in feature space – high correlation (desirable) produces a 'low' curve whereas low correlation produces a 'high' curve.

[1] Using *projected* joint centres rather than their full 3D position avoids many (though not all) problems associated with 'kinematic flip' ambiguities [11] where very different poses give rise to very similar projected joint centres.

Figure 6: Four test datasets: (a) clean silhouettes; (b) with added noise; (c) with lower quarter removed; (d) real silhouettes manifesting some segmentation error.

4 Results

We compared the three selected shape descriptors using four test datasets (Figure 6) containing silhouettes that were: (i) perfect; (ii) noisy; (iii) partially occluded; (iv) real.

We begin by comparing the three methods for clean data (Figure 6a) taken directly from the synthetic dataset. Figure 7a shows that, although Lipschitz embeddings perform slightly worse than the other descriptors, accuracy is similar for all three representations.

To create a noisy data-set, we corrupted the clean test silhouettes with Gaussian noise along the contour (Figure 6b). Such corruption typically results from segmentation errors at the boundaries and compression artefacts. From Figure 7b, we see that performance is largely unchanged by the added noise, with the exception that DCT coefficients marginally outperform the Histogram of Shape Contexts. This may be explained by the fact that lower order DCT coefficients (as used in this case) encode the lower frequencies within the image and therefore suppress noise. Again, Lipschitz embeddings do not perform as well as the other two methods.

In order to simulate occluded data, we removed the bottom quarter of each test silhouette and renormalized, as if the subject had been obscured from approximately knee-level down (Figure 6c). Although this is a relatively crude approach, it presents each method with data that is somewhat different from the training data yet is typical in real life applications. Figure 7c shows that the Histogram of Shape Contexts performs well for small k (approximately the top 1% of the data) but is out-performed for higher k by the DCT. Lipschitz embeddings are again typically out-performed by the other two methods.

For the final experiment, we use real silhouettes from a 'starjumps' sequence (Figure 6d), obtained via background subtraction and with projected joint centres labelled by hand. Due to the limited number of test images, the curves in Figure 7d are slightly noisier but suggest that DCT coefficients significantly outperform both Histogram of Shape Contexts and Lipschitz embeddings. More specifically, the Histogram of Shape Contexts and Lipschitz embeddings have perform similar to a random ranking for this data-set. This is a surprising and interesting result, particularly since this is arguably the most important test set of the four. It may be questioned whether the normalization procedure employed in this experiment might favour one method over another. However, the test silhouettes show little corruption that would have a significant effect on this process.

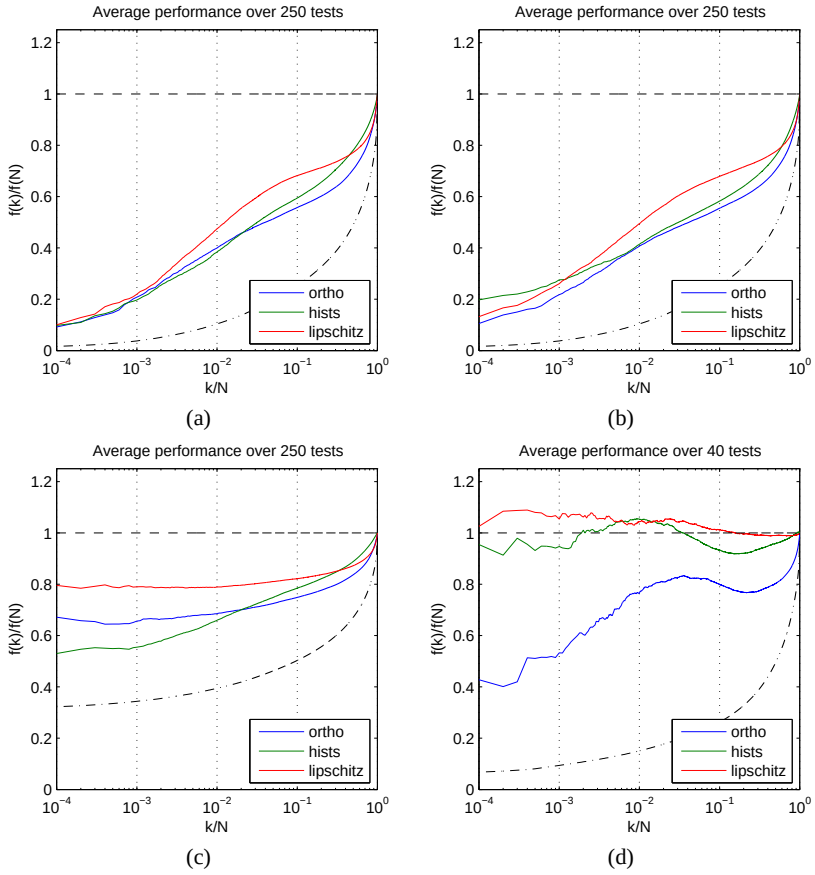


Figure 7: Results for (a) clean data; (b) noisy data; (c) occluded data; (d) real data. Curves correspond to DCT coefficients (*ortho*), Histogram of Shape Contexts (*hists*) and Lipschitz embeddings (*lipschitz*)

5 Conclusion

We have presented a comparison of three shape descriptors for the application of human pose estimation from binary silhouettes. In particular, we compare two straightforward and established methods (the DCT and Lipschitz embeddings) against the recently proposed Histogram of Shape Contexts (HoSC), a 'local' descriptor that is claimed to be superior to 'global' methods. However, despite its computational complexity, our results suggest that the HoSC offers little (if any) benefit over the alternative, simpler methods.

Although it has not escaped our attention that some of our results appear to contradict those that have appeared in previous works, we note that these studies often employed a limited number of training images [1] or more a complex matching process [2]. To the best of our knowledge, this study is the first to evaluate such descriptors under controlled conditions where meaningful comparisons can be made.

References

[1] A. Agarwal and B. Triggs. Recovering 3D human pose from monocular images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(1):1–15, January 2006.

[2] V. Athitsos, J. Alon, S. Sclaroff, and G. Kollios. BoostMap : A method for efficient approximate similarity rankings. In *Proc. 22nd IEEE Conf. on Comp. Vis. and Patt. Rec.*, volume 2, pages 268–275, 2004.

[3] V. Athitsos and S. Sclaroff. Estimating 3D hand pose from a cluttered image. In *Proc. 21st IEEE Conf. on Comp. Vis. and Patt. Rec.*, volume 2, pages 432–442, 2003.

[4] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):509–522, April 2002.

[5] G. R. Hjaltason and H. Samet. Properties of embedding methods for similarity searching in metric spaces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(5):530–549, May 2003.

[6] M. K. Hu. Visual pattern recognition by moment invariants. *IRE Trans. Inform. Theory*, 8:179–187, February 1962.

[7] L. J. Latecki and R. Lakamper. Convexity rule for shape decomposition based on discrete contour evolution. *Comput. Vis. Image Und.*, 73(3):441–454, March 1999.

[8] R. Mukundan, S. H. Ong, and P. A. Lee. Image analysis by Tchebichef moments. *IEEE Trans. Image Process.*, 10(9):1357–1364, September 2001.

[9] K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, and S. W. Zucker. Shock graphs and shape matching. *Int. J. Comput. Vis.*, 35(1):13–32, November 1999.

[10] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Discriminative density propagation for 3D human motion estimation. In *Proc. 23nd IEEE Conf. on Comp. Vis. and Patt. Rec.*, volume 1, pages 390–397, 2005.

[11] C. Sminchisescu and B. Triggs. Kinematic jump processes for monocular 3D human tracking. In *Proc. 21st IEEE Conf. on Comp. Vis. and Patt. Rec.*, volume 1, pages 69–76, 2003.

[12] M. R. Teague. Image analysis via the general theory of moments. *J. Opt. Soc. Am.*, 70:920–930, August 1980.

[13] P. Tresadern. *Visual Analysis of Articulated Motion*. PhD thesis, University of Oxford, October 2006.

[14] P.-T. Yap, R. Paramesran, and S.-H. Ong. Image analysis by Krawtchouk moments. *IEEE Trans. Image Process.*, 12(11):1367–1377, November 2003.

[15] D. S. Zhang and G. Lu. A comparative study of curvature scale space and Fourier descriptors. *J. Vis. Commun. Image R.*, 14(1):41–60, March 2003.

[16] D. S. Zhang and G. Lu. Study and evaluation of different Fourier methods for image retrieval. *Image Vision Comput.*, 23(1):33–49, January 2005.

Layered image model using binary PCA transparency masks

Zoran Zivkovic
ISLA Lab, University of Amsterdam, The Netherlands
zivkovic@science.uva.nl

Abstract

The "layered image model" [13] represents an image sequence as a composition of 2D layers where each layer corresponds to a different object. A layer is described by its appearance and its transparency mask. The transparency masks are used to combine the layers. In this paper we present a probabilistic layered model that uses the "logistic principal component analysis (PCA)" to describe the masks. The Gaussian based factor analysis was used previously but it does not consider the constraints imposed on the transparency values. The "logistic PCA" models the transparency values that are between 0 and 1 more naturally using Bernoulli distributions. The presented model can be used to automatically extract low dimensional representation of the transparency maps of the moving objects from a video sequences more efficiently.

1 Introduction

In the layered representation [13] a video sequence of a 3D scene is decomposed into a set of 2D layers where each layer corresponds to a different moving object. This is a potentially very effective representation for automatically analyzing video sequences since the representation greatly simplifies the geometry but still accounts for the occlusions between the layers [8].

A generative probabilistic layered image model is presented by Jojic and Frey [8] and further extended by a number of authors. Each layer in the layered model is described by its appearance and its transparency mask. The sprite appearances are combined using the transparency masks. Various appearance models were proposed: Gaussian per pixel [8], factor analysis [6], index maps [7], Gaussian with local image deformations [9] etc. Various models were also proposed for the transparency maps: Gaussian [8], factor analysis [6], binary mask with local image deformations [9] etc.

Principal component analysis (PCA) and factor analysis (FA) are often used for modelling image data [12, 1]. Both techniques try to find a low dimensional representation of the data by linear projection. Layered model presented by Frey et. al. [6] is using factor analysis for layer appearance and the transparency map. The model can be used to automatically extract low dimensional representation of the moving objects from a video sequence. For example, images of a person walking can be mapped to a 1-dimensional manifold that measures the phase of the persons gait. The FA can be applied to find the low dimensional representation of both the layer transparency masks and the layer appearance. However, the Gaussian based factor analysis does not consider the constraints

imposed on the transparency values. Furthermore, a number of authors noted that more efficient and robust inference can be achieved by using Bernoulli distribution instead of Gaussian for the transparency masks [14, 9].

The relation between the low dimensional representations of the mask and the appearance is complex in general. For example a single-colored object might change its transparency mask, 2D shape, while the appearance remains the same. Therefore in this paper we leave the choice of the appearance model free and focus on the low dimensional representation of the transparency masks. Natural model for the masks is to use the Bernoulli distribution [14, 9]. The "logistic PCA" using Bernoulli distributions was proposed in the machine learning community [11, 10]. The recent study [16] shows that the "binary PCA" is much more accurate than the standard PCA in representing binary image data and probability maps. In this paper we will present a layered model where the "logistic PCA" is used for the transparency masks. The presented model can be used to automatically extract low dimensional representation of the moving objects transparency mask (shape) from a video sequence more efficiently and robustly.

This paper is organized as follows. In Section 2 we describe the layered image model. In Section 3 the model is extended by including the "logistic PCA" transparency masks. In Sections 4 we explain a generalized expectation maximization (EM) inference scheme for the model. In Section 5 we present experimental results, and in Section 6 we list our conclusions and some topics for further research.

2 Layered image model

In the layered model an image $\mathbf{x}$ is decomposed into a set of L layers corresponding to objects that occlude each other. Each layer is described by its appearance parameters Λ_l and the transparency mask $\mathbf{m}_l$.

The transparency mask describes which part of the image is covered by the object. The mask value for the l-th layer and d-th pixel will be denoted by $\mathbf{m}_{dl} \in \{0,1\}$. A natural way to model the mask data is using Bernoulli distributions:

$$p(\mathbf{m}_{dl}|\alpha_l) = {\alpha_{dl}}^{\mathbf{m}_{dl}} \bar{\alpha}_{dl}^{\bar{\mathbf{m}}_{dl}} \tag{1}$$

where α_{dl} is the probability that $\mathbf{m}_{dl} = 1$, $\bar{\mathbf{m}}_{dl} = 1 - \mathbf{m}_{dl}$ and $\bar{\alpha}_{dl} = 1 - \alpha_{dl}$.

The appearance model describes the pixel values. The probability of the d-th image pixel value $\mathbf{x}_d$ for the l-th layer is given by $p(\mathbf{x}_d; \Lambda_l)$. The pixel value $\mathbf{x}_d$ is for example the 3 dimensional RGB value. In this paper we will use a simple appearance model, similar to [8], consisting of a Gaussian per pixel

$$p(\mathbf{x}_d; \Lambda_l) = \mathcal{N}(\mathbf{x}_d; \mu_{dl}, \Psi_{dl} I) \tag{2}$$

where the covariance matrix is isotropic $\Psi_{dl} I$ and I is a 3×3 identity matrix.

Assuming the pixel values to be independent an image is described by:

$$\prod_d p(\mathbf{x}_d, \mathbf{m}_{d1}, ..., \mathbf{m}_{dL}|\Omega) \tag{3}$$

where $\Omega = \{\Lambda_1, ..., \Lambda_L, \alpha_1, ..., \alpha_L\}$ are the parameters of the model. The unobserved mask variables $\mathbf{m}_{d1}, ..\mathbf{m}_{dL}$ determine which pixels belong to which objects/layers as described further.

To alow the objects to switch between layers an additional discrete labelling variable c needs to be included which assigns objects to different layers, see [8]. For simplicity here we will assume that each object stays always in the same layer.

The layered model per pixel $p(\mathbf{x}_d, \mathbf{m}_{d1}, ..., \mathbf{m}_{dL}|\Omega)$ can be written using recursive equation:

$$p(\mathbf{x}_d, \mathbf{m}_{dl}, ..., \mathbf{m}_{dL}, \mathbf{o}_{dl}) = p(\mathbf{x}_d; \Lambda_l)^{\mathbf{m}_{dl}\mathbf{o}_{dl}} p(\mathbf{x}_d, \mathbf{m}_{dl+1}, ..., \mathbf{m}_{dL}, \mathbf{o}_{dl+1}) p(\mathbf{m}_{dl}|\alpha_l)^{\mathbf{o}_{dl}} \quad (4)$$

where $\mathbf{o}_{dl} = \prod_1^{l-1} \bar{\mathbf{m}}_{dl}$ is the occlusion of the d-th pixel by the previous layers closer to the camera. The equation describes stacking the layers on top of each other with background layer $l = L$ at the bottom. In other words a pixel value $\mathbf{x}_d$ can be explained by the l-th layer appearance model $p(\mathbf{x}_d; \Lambda_l)$ if it is not occluded and already modelled by the previous layers $1, ..., l-1$, i.e. $\mathbf{o}_{dl} = 0$, and if the current layer mask $\mathbf{m}_{dl} = 1$. If the pixel value is not described by the current and the previous layers, i.e. $\mathbf{o}_{dl} = 0$ and $\mathbf{m}_{dl} = 0$, than it is described by the layers that lie below $p(\mathbf{x}_d, \mathbf{m}_{dl+1}, ..., \mathbf{m}_{dL}, \mathbf{o}_{dl})$. For simplicity the background layer $l = L$ will be without the mask $p(\mathbf{x}_d, \mathbf{m}_{dL}, \mathbf{o}_{dl}) = p(\mathbf{x}_d; \Lambda_L)^{\mathbf{o}_{dL}}$. For the top layer there are no occlusions $p(\mathbf{x}_d, \mathbf{m}_{dl}, ..., \mathbf{m}_{dL}|\Omega) = p(\mathbf{x}_d, \mathbf{m}_{d1}, ..., \mathbf{m}_{dL}, \mathbf{o}_{dl-1} \equiv 0)$.

A common extension is to include unknown layer transformation function $\mathscr{T}_{tl}$, for example translation, rotation, scaling etc. The transformation $\mathscr{T}_{tl}$ transforms the layer l before it is combined with other images. We will denote the transformed appearance parameters by $\mathscr{T}_{tl}\Lambda_l$ and the transformed mask parameters as $\mathscr{T}_{tl}\alpha_l$. The recursive equation per pixel and per layer becomes:

$$p(\mathbf{x}_d, \mathbf{m}_{dl}, ..., \mathbf{m}_{dL}, \mathbf{o}_{dl}) =$$
$$p(\mathbf{x}_d; \mathscr{T}_{tl}\Lambda_l)^{\mathbf{m}_{dl}\mathbf{o}_{dl}} p(\mathbf{x}_d, \mathbf{m}_{dl+1}, ..., \mathbf{m}_{dL}, \mathbf{o}_{dl+1}) p(\mathbf{m}_{dl}|\mathscr{T}_{tl}\alpha_l)^{\mathbf{o}_{dl}} p(\mathscr{T}_{tl}) \quad (5)$$

where $\mathscr{T}_{t1}, ..., \mathscr{T}_{tL}$ are additional unobserved variables. As in [8] we consider a discrete set of transformations $\mathscr{T}_{tl} \in \{\mathscr{T}_1, ..., \mathscr{T}_T\}$ and the prior distribution over the transformations is denoted as $p(\mathscr{T}_{tl}) = p_{tl}$.

3 Logistic PCA masks

We would like to design a layered image model to automatically extract low dimensional representation of the moving objects transparency masks. For example, images of a person walking can be mapped to a 1-dimensional manifold that measures the phase of the persons gait, see Figure 1. Principal component analysis (PCA) commonly used to find a low dimensional representation of the data by linear projection. We describe here a similar model but for Bernoulli distributions, the so called "logistic PCA", to describe the transparency masks of the layered model from the previous section. As in [10] instead of the α_{dl} in (1) for the Bernoulli mask model we will use the log-odds parameter $\Theta_{dl} = \log(\alpha_{dl}/(1-\alpha_{dl}))$ and the logistic function $\sigma(\Theta_{dl}) = (1+e^{-\Theta_{dl}})^{-1}$. The mask model can be written equivalently as:

$$p(\mathbf{m}_{dl}|\Theta_{dl}) = \sigma(\Theta_{dl})^{\mathbf{m}_{dl}} \sigma(-\Theta_{dl})^{\bar{\mathbf{m}}_{dl}} \quad (6)$$

Logistic PCA assumes that the log-odds mask parameter Θ_l is given by the so called "mean" log-odds mask parameter Δ_l plus a linear combination of $S \ll D$ basis vectors

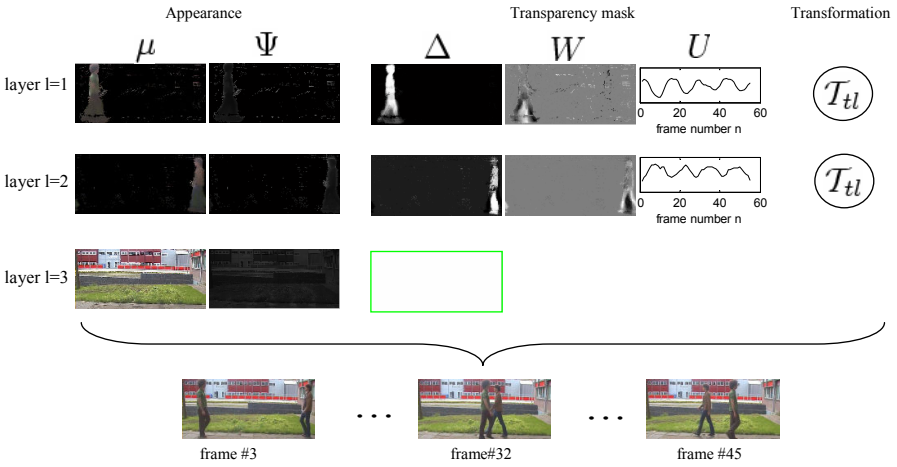


Figure 1: The parameters of the layered model learned automatically from the image sequence of two people walking in opposite directions and occluding each other.

(images) contained in the rows of the $S \times D$ matrix W_l. The linear combination is obtained through the coefficients contained in U_l:

$$\Theta_{dl} = \Delta_{dl} + \sum_s U_{sl} W_{sdl}. \tag{7}$$

The Δ_l,W_l and U_l are the parameters of the logistic PCA.

4 Learning model parameters

The layered image model with the binary PCA for a set of N independent images can be written as:

$$\prod_{nd} p(\mathbf{x}_{nd}, \mathbf{m}_{nd1}, ..., \mathbf{m}_{ndL}, \mathcal{T}_{nt1}, ..., \mathcal{T}_{ntL} | \Omega) \tag{8}$$

where the masks $\mathbf{m}_{nd1}, ..\mathbf{m}_{ndL}$ and the transformations $\mathcal{T}_{nt1}, ..., \mathcal{T}_{ntL}$ are the unobserved variables and $\Omega = \{\Lambda_1, ..., \Lambda_L, \Delta_l, ..., \Delta_L, W_1, ..., W_L, U_1, ..., U_L, p_{tl}\}$ are the parameters of the model. The index n indicates that there is each layer can have a different appearance mask $\mathbf{m}_{ndl}$ and a different transformation $\mathcal{T}_{ntl}$ for each image. The log-likelihood of a given set of N images is given by:

$$\mathcal{L}(\Omega) = \sum_{nd} \ln p(\mathbf{x}_{nd} | \Omega) \tag{9}$$

where the unknown masks and transformations are integrated out:

$$p(\mathbf{x}_{nd} | \Omega) = \sum_{\text{all masks and transf.}} p(\mathbf{x}_{nd}, \mathbf{m}_{nd1}, ..., \mathbf{m}_{ndL}, \mathcal{T}_{nt1}, ..., \mathcal{T}_{ntL} | \Omega) \tag{10}$$

The goal is to find the parameters Ω that maximize the log-likelihood (9).

4.1 Approximate inference

The log-likelihood is a complex function. The EM algorithm [3] presents an iterative solution but computing it would be intractable for such a model [8]. Therefore as in [8] we use a variational approximate method. We will denote the hidden variables by h, layer masks and hidden transformations in our case. Variational techniques replace the intractable computation of the posterior distribution $p(h|\mathbf{x})$ with a search for a simplified distribution $q(h)$, that is made close to $p(h|\mathbf{x})$ by minimizing the "free energy" function:

$$F = \int_h q(h) \frac{p(h)}{\ln p(\mathbf{x}, h|\Omega)} \geq -\mathscr{L}(\Omega) \tag{11}$$

Minimizing F w.r.t. $q(h)$ minimizes the relative entropy between $q(h)$ and $p(h|\mathbf{x})$. Minimizing F w.r.t. $q(h)$ and the model parameters Ω minimizes an upper bound on the negative log-likelihood of the data $\mathscr{L}(\Omega)$ [5].

Similar to [8] we will use the following simplified factorized distribution:

$$q(h) = \prod_{ndl} \mathbf{r}_{ndl}{}^{\mathbf{m}_{ndl}} \bar{\mathbf{r}}_{ndl}^{\bar{\mathbf{m}}_{ndl}} \mathbf{q}_{ntl} \tag{12}$$

The parameter estimation is then performed iteratively using a generalized EM algorithm steps:

E step:	Minimizing F w.r.t. the variational	(13)
	parameters $\mathbf{r}_{ndl}$ and $\mathbf{q}_{ntl}$.	(14)
M step:	Minimizing F w.r.t. Ω.	(15)

These two steps are repeated iteratively until convergence. See [5] for a tutorial.

4.2 Updating mask parameters

The update equations for the E and M steps above are already given in the various extensions [6, 9] of the initial work by Jojic and Frey [8]. An extensive tutorial can be found also in [5]. Therefore, and because of the limited space, we will not repeat all the update equations. Instead we will focus on the extension proposed in this paper: the logistic PCA model applied to the transparency masks and the update equations for the logistic PCA parameters Δ, W and U.

The variational parameters are updated in the E-step. The layer appearance parameters are updated in the M-step. In the M step we need also to minimize the free energy function F w.r.t. the logistic PCA parameters Δ, W and U. It can be shown, see (5) and (12), that the only part of the function F that depends on the layer l mask parameters is given by:

$$\sum_{n,t,d} \mathbf{q}_{ntl} (\mathbf{w}_{dl} \mathbf{r}_{ndl} \log(\mathscr{T}_{ntl} \alpha_{dl}) + \mathbf{w}_{dl} \bar{\mathbf{r}}_{ndl} \log(\mathscr{T}_{ntl} \bar{\alpha}_{dl})) \tag{16}$$

where $\mathbf{w}_{dl} = \prod_1^{l-1} \bar{\mathbf{r}}_{ndl}$. So in order to minimize the free energy function F w.r.t. the logistic PCA parameters Δ, W and U for each layer we need to consider only these terms.

For simplicity as previously in the similar models [4] we assume the transformation $\mathscr{T}_{ntl}$ to be a permutation matrix that rearranges the pixels. For example, to account for all translations in a $J \times J$ image, $\mathscr{T}_{ntl}$ can take on J^2 values the permutation matrices that account for all discrete translations. The discrete 2D image translation is a common transformation to align images. Furthermore, an efficient solution for the E step is available [4]. Furthermore, note that by transforming an image to log-polar coordinates, shifts correspond to rotations and scalings [15]. Let $\mathscr{T}_{ntl}^{-1}$ denote the inverse transformation. The terms above (16) can be rewritten in the following form:

$$\sum_{n,t,d} \mathbf{q}_{ntl}(\mathscr{T}_{ntl}^{-1}\mathbf{w}_{dl}\mathbf{r}_{ndl}\log(\alpha_{dl}) + \mathscr{T}_{ntl}^{-1}\mathbf{w}_{dl}\bar{\mathbf{r}}_{ndl}\log(\bar{\alpha}_{dl})) \tag{17}$$

After integrating over all possible transformations we get:

$$\sum_{n,d} \widehat{\mathbf{wr}_{ndl}}\log(\alpha_{dl}) + \widehat{\mathbf{w}\bar{\mathbf{r}}_{dl}}\log(\bar{\alpha}_{dl}) \tag{18}$$

where:

$$\widehat{\mathbf{wr}_{ndl}} = \sum_t \mathbf{q}_{ntl}\mathscr{T}_{ntl}^{-1}\mathbf{w}_{dl}\mathbf{r}_{ndl} \tag{19}$$

$$\widehat{\mathbf{w}\bar{\mathbf{r}}_{dl}} = \sum_t \mathbf{q}_{ntl}\mathscr{T}_{ntl}^{-1}\mathbf{w}_{dl}\bar{\mathbf{r}}_{ndl}. \tag{20}$$

Finally this can be written using the log-odds parameters $\Theta_{dl} = \log(\alpha_{dl}/(1-\alpha_{dl}))$ as:

$$\omega_{ndl}(\widehat{M_{ndl}}\Theta_{ndl} + \log\sigma(-\Theta_{ndl})) \tag{21}$$

where

$$\omega_{ndl} = (\widehat{\mathbf{wr}_{ndl}} + \widehat{\mathbf{w}\bar{\mathbf{r}}_{dl}}) \text{ and} \tag{22}$$

$$\widehat{M_{ndl}} = \widehat{\mathbf{wr}_{ndl}}/\omega_{ndl}. \tag{23}$$

Note that $\widehat{M_{ndl}} \in [0,1]$. If we consider $\widehat{M_{ndl}}$ as data then (21) presents log-likelihood under Bernoulli model with the log-odds parameters Θ_{ndl}. Additionally each data point is weighted by its weight ω_{ndl}. The goal in the M-step is to find the logistic PCA parameters Δ_l, W_l and U_l that maximize the weighted log-likelihood (21). The maximum can not be found in closed form. There exist an efficient iterative procedure for the logistic PCA [10]. The procedure can be extended for the weighted case (21). For completeness of the text the iterative update equations for the weighted logistic PCA are given in Appendix A.

4.3 Practical algorithm

For the sake of clarity we summarize the practical algorithm:

Initialization: In case of a static background the background layer appearance parameters can be initialized by the mean value and the variance of each pixel for the whole sequence. The other layer appearances can be initialized by arbitrary mean and some

large variance. For the masks we initialize the logistic parameter Δ_l by some small random values around zero. Within first few iterations it is useful to update the parameters for each layer separately starting from the background layer and going upwards. This is similar to the greedy layer estimation presented in [14]. Furthermore, for the first few iterations we keep the basis vectors of the logistic PCA W and the coefficients U to zero and then initialize them by some small random values, for example sampled from a zero mean Gaussian distribution with the standard deviation 0.001.

1: For each image calculate the approximation of the posterior distribution by maximizing F w.r.t. the variational parameters $\mathbf{r}_{ndl}$ and $\mathbf{q}_{ntl}$, see [5].

2: For each image calculate ω_{ndl} and $\widehat{M_{ndl}}$.

3: Update the appearance parameters Λ_l and if required p_t.

4: Update the logistic PCA mask parameter estimates Δ_l, W_l and U_l using the update equations from Appendix A. The updated Δ_l, W_l and U_l will not maximize the free energy function F but they will increase its value. This can be seen as a "generalized M step" [5].

5: Stop if increase of the free energy function F is below some threshold, otherwise go to **1**.

There is often not enough data to estimate p_t reliably and we will use in this paper a uniform prior distribution over the transformations: $p_t = 1/D$ [8].

5 Experiments

5.1 Extracting low dimensional representations

To demonstrate how the layered model can automatically extract low dimensional representation of the transparency maps of the moving objects from a video sequences we recorded a 55 frame sequence of two people walking into opposite directions and occluding each other during the sequence. In Figure 1 we present the model parameters automatically learned from the sequence. Note that U nicely captures the cyclical walking motion while W models the corresponding deformations.

5.2 Reconstructing transparency maps

In order to compare the quality of the Gaussian and Bernoulli based models we conducted the following experiments. We used a sequence captured by a surveillance camera. The camera was observing people walking in front of a static background. The sequence contains 400 frames. There were 5 people present in the sequence. Only single person was present per frame. Therefore we constructed a model consisting of 2 layers. The appearance of the front layer was modelled by a Gaussian mixture with 5 components to accommodate for the 5 different people. The transparency mask is modelled by 5 component logistic PCA. The parameters learned from the sequence are presented in Figure 2. The learned Gaussian mixture components nicely correspond to the 5 different people present in the video. The components of the logistic PCA presented at the bottom row in Figure 2 seem to capture the walking deformations and also the different walking directions. The first 2 people were observed walking in both directions and this is clearly visible in their appearance parameters.

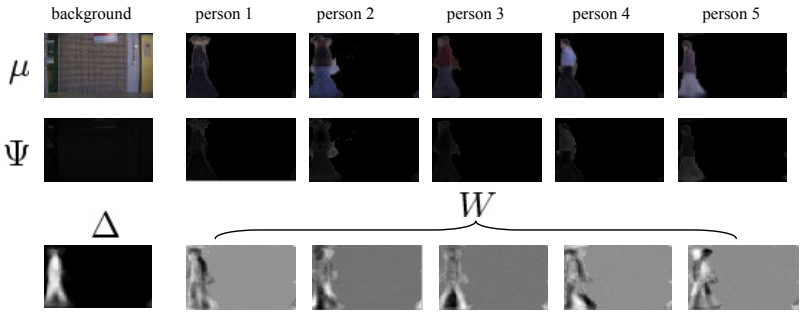


Figure 2: The parameters of the model learned from a 400 frames sequence containing 5 walking people. Only single person was present per frame. The appearances of the 5 people and the static background are presented at the top rows. The first 2 people were observed walking in both directions. At the bottom we present the parameters of the logistic PCA ($S = 5$ components) that is used to model the transparency maps segmenting the people from the background. The mean and the 5 basis images are shown.

Once the model is constructed we used additional 50 images to test the quality of the model. The ground truth segmentation of the 50 additional images is obtained by manually segmenting the persons from the background. Using the model, we compress the images to the PCA scores for the transparency masks. We use then the model to project the PCA scores back to mask images. Finally, we use most likely transformation $argmax(\mathbf{q}_{ntl})$ to shift the reconstructed mask to the proper position. The mask reconstructed by the layered model will be denoted $\hat{X}_n$. See some examples in Table 5.2. We then measure the difference between the manually segmented image and the segmentation using the layered model. We measured the error in three ways. (i) Quadratic loss: the sum of the squared differences per pixel value, $e_2 = (1/D)\sum_d (X_{nd} - \hat{X}_{nd})^2$. (ii)logistic loss: the sum of the log-likelihood of the ground truth masks given the reconstructions, $e_{log} = 1/D \sum_d X_{nd} \ln \hat{X}_{nd} + (1 - X_{nd}) \ln(1 - \hat{X}_{nd})$. As the reconstruction from the Gaussian model can be outside $(0,1)$, we first map values outside this interval to $\varepsilon = 10^{-6}$ and $1 - epsilon$ respectively.(iii) Zero-one loss: first we threshold the segmentation by the model at $\hat{X}_{nd} > 1/2$ to get a binary reconstruction $\hat{X}^{01}_n$, then we measure the number of pixels that differ from the ground truth, $e_{01} = (1/D)\sum_d |X_{nd} - \hat{X}^{01}_{nd}|$.

The results for $S = 5$ components are reported in Table 5.2. We also constructed a layered model similar to [6] where we used the Gaussian probabilistic PCA to model the transparency masks. Clearly, the layered model using logistic PCA leads to big improvements. This is also visible in Figure 5.2.

Since the camera was static, in Table 5.2 we also show the results obtained using standard background subtraction scheme [2] which builds a model only for the background layer. The layered model which considers all layers leads to much better results, see Table 5.2. Another common technique to improve segmentation results after background subtraction is to apply some morphological operators on the segmentation results. We used image closing operator with a 3×3 element. The results improve slightly but the layered model is still superior. When a larger template is used for image closing of when image opening is performed, the results get only worse.

original	
manual	
background subtraction [2]	
background subtraction + image closing	
layered model + normal PCA	
layered model + logistic PCA	

Table 1: Segmentation examples for various approaches.

	e_2	e_{01}	e_{log}
layered model + logistic PCA ($S = 5$ components)	0.023 (0.006)	0.03 (0.01)	0.086 (0.023)
layered model + norm. PCA ($S = 5$ components)	0.029 (0.006)	0.04 (0.01)	0.098 (0.023)
background subtraction + image closing	0.052 (0.017)	0.052 (0.017)	0.64 (0.23)
background subtraction	0.059 (0.024)	0.059 (0.024)	0.66 (0.28)

Table 2: Segmentation error w.r.t. manually segmented images of the walking people sequence. The mean error per pixel over 50 hand-segmented images is reported. The standard deviation over images is reported within the brackets.

6 Conclusions

The generative probabilistic layered image model presented by Jojic and Frey [8] was further extended by a number of authors. We focus here on modelling the transparency masks. The natural way to model the masks is to use Bernoulli distributions. We presented probabilistic layered image model that models the masks using Bernoulli distributions and extracts the low dimensional representation of the transparency masks using the "logistic PCA" [10]. Gaussian component and factor analysis as in [6] does not take into account that the transparency mask has values limited between [0,1]. The logistic PCA describes the mask more naturally and leads to crisper masks and better segmentation results as we demonstrated.

A disadvantage of the logistic PCA is that it requires solving two $S \times S$ linear systems for each data point (see Appendix) which might be prohibitive if the number of components S is large. Furthermore, projecting data to the low-dimensional PCA space requires iterations in the case of logistic PCA, while for normal PCA the projection is linear. Finally, the logistic PCA used here is not a full generative model as there is no prior distribution on the low-dimensional coefficient matrix U. A computationally slightly more expensive model which incorporates Gaussian priors on U is described in [11].

Appendix I: Weighted Binary PCA update equations

U-update: First intermediate quantities are computed:

$$H_{nd} = \omega_{nd}\Theta_{nd}^{-1}\tanh(\Theta_{nd}/2)\ , A_{nss'} = \Sigma_d H_{nd}W_{sd}W_{s'd} \text{ and} \tag{24}$$

$$B_{ns} = \Sigma_d(2\omega_{nd}\widehat{M}_{nd} - 1 - H_{nd}\mu_d)W_{sd} \tag{25}$$

Row n of U is computed by solving linear system: $\Sigma_{s'}A_{nss'}U_{ns'} = B_{ns}$.

W-update: First intermediate quantities are computed:

$$A_{dss'} = \Sigma_n H_{nd}U_{ns}U_{ns'} \text{ and } B_{ds} = \Sigma_n(2\omega_{nd}\widehat{M}_{nd} - 1 - H_{nd}\mu_d)U_{ns} \tag{26}$$

Column d of W is computed by solving the linear system $\Sigma_{s'}A_{dss'}W_{s'd} = B_{dl}$.

$$\Delta-\textbf{update}:\Delta_{nd} = (\Sigma_n H_{nd})^{-1}\Sigma_n(2\omega_{nd}\widehat{M}_{nd} - 1 - H_{nd}(UV)_{nd}) \tag{27}$$

References

[1] M. Black and A. Jepson. EigenTracking: Robust matching and tracking of articulated objects using a view-based representation. *Int. Journal of Computer Vision*, 26(1):63–84, 1998.

[2] C.Stauffer and W.Grimson. Adaptive background mixture models for real-time tracking. *In Proc. of the Conf. on Computer Vision and Pattern Recognition*, 1999.

[3] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Stat. Society, Series B (Methodological)*, 1(39):1–38, 1977.

[4] B. Frey and N. Jojic. Transformation-invariant clustering using the EM algorithm. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25:1–17, 2003.

[5] B. Frey and N. Jojic. A comparison of algorithms for inference and learning in probabilistic graphical models. *IEEE Trans. on Pattern Analysis and Machine Intel.*, 27(9), 2005.

[6] B. Frey, N. Jojic, and A. Kannan. Layered density models and unsupervised video analysis. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2003.

[7] N. Jojic and Y. Caspi. Capturing image structure with probabilistic index maps. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2004.

[8] N. Jojic and B. Frey. Learning flexible sprites in video layers. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2001.

[9] A. Kannan, N. Jojic, and B. Frey. Layers of appearance and deformation. *10th Int. Workshop on Artificial Intelligence and Statistics (AISTATS)*, 2005.

[10] A. Schein, L. Saul, and L. Ungar. A generalized linear model for principal component analysis of binary data. *In Proc. Int. Workshop on Art. Intel. and Statistics*, pages 14–21, 2003.

[11] M. Tipping. Probabilistic visualisation of high-dimensional binary data. *In Advances in Neural Information Processing Systems (NIPS)*, 1999.

[12] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3:71–86, 1991.

[13] J. Wang and E. Adelson. Layered representation for motion analysis. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 361–366, 1993.

[14] C. Williams and M. Titsias. Learning about multiple objects in images: Factorial learning without factorial search. *In Advances in Neural Information Processing Systems (NIPS)*, 2003.

[15] G. Wolberg and S. Zokai. Robust image registration using log-polar transform. *In Proc. IEEE Int. Conf. image processing, vol. 1*, pages 493–496, 2000.

[16] Z.Zivkovic and J. Verbeek. Transformation invariant component analysis for binary images. *In Proc.IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

Isolating Motion and Color in a Motion Blurred Image

Alessandro Giusti Vincenzo Caglioti
Dipartimento di Elettronica e Informazione, Politecnico di Milano
P.za Leonardo da Vinci, 32
20133 Milano - Italy
{giusti,caglioti}@elet.polimi.it

Abstract

Photographic images of moving objects are often characterized by motion blur; analyzing motion blurred images is problematic since the moving object boundaries appear fuzzy and seamlessly blend with the background. In extreme cases, when the object motion is fast in relation to the exposure time, the blurred object image becomes an elongated, semitransparent smear.

We consider a motion-blurred color image of an object moving over a still background: we introduce meaningful entities, the "alpha map" and the "color map", which bear information about the object motion during the exposure, and its color and texture; we draw connections to the well-known alpha matting problem, providing an original interpretation in this context; we present an analytic technique for extracting the two maps under assumptions on the background and object colors, and explore the relaxation of these assumptions. We provide experimental results on both synthetic and real images, which confirm the correctness of our approach, and describe diverse application examples in fields spanning from 3D reconstruction to image/video enhancement.

1 Introduction

When a moving object is photographed by a still camera, its image is motion blurred because its position changes during the exposure time. If the exposure time is not short enough in relation to the object apparent speed, the object results in a visible "smear" or "streak" in the image, and its contours blend with the background confusing traditional computer vision techniques.

In this paper, the blurred image is analyzed by producing two pieces of information for each image pixel:

- how long the moving object projected there during the exposure time ($\alpha(p)$);
- the color the pixel would have if the background was black ($o(p)$).

Over the whole image, they comprise an "alpha map" and a "color map" (see figure 1). We show that this representation is strictly related to the deeply-studied *alpha matting* problem.

We describe an analytical procedure for computing such maps, in a single blurred color image of a monochromatic object moving over a known background. A monochromatic object is an object whose surface is pigmented with a single color and, possibly, its darker shades – including black; we also extend the technique in order to handle bichromatic objects, pigmented with two different colors, their combinations and darker shades; many common objects, including most types of sport balls can be considered either monochromatic or bichromatic. We show that any further relaxation of the assumptions makes the problem underconstrained: we are considering the most general scenario in which the problem has an analytical solution.

We maintain that, once known, the alpha and color maps can be a valuable aid in the understanding of the blurred object image. Other than applications for image enhancement – for example to highlight images of fast-moving objects (see section 4 and additional material), or to recover the color of a very blurred monochromatic object, the alpha map can be exploited for reconstructing the fast motion of objects which appear blurred, or as a necessary preprocessing step for precisely deblurring a moving object on a sharp background; actually, many works exploiting motion blur of objects as a source of information would benefit of our provided separation of background, object exposure time and object color/texture. In section 5, we briefly explain some of our current work in the field.

The paper is structured as follows: after referencing related works, we formally define the problem of reconstructing the motion blur of a moving object over a known background and highlight analogies to the alpha matting problem (section 2); we provide a theoretically founded, analytical solution to the problem under the assumption that the object is monochromatic, then we explore extensions and limits of the technique (section 3); we then show several practical applications and experimental results (section 4) and finally draw conclusions and outline future developments (section 5).

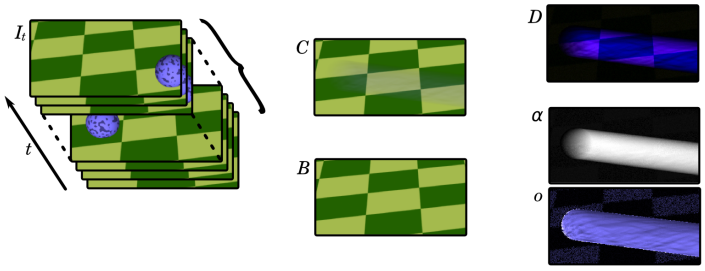


Figure 1: The image with the (barely visible) motion blurred object (C, at the center) can be interpreted as the temporal average over the exposure time of infinite still images I_t (left). Ordinary background subtraction (image D, after equalization) does not help in characterizing the blur. Our technique provides a map α (analogous to an alpha matte) summarizing the object motion and an object color map o.

1.1 Related Works

Motion blur analysis has been exploited for a number of applications: for example, in [14, 15] quantitative measurements of blur are used in order to estimate the speed of vehi-

cles and spherical objects: the blur parameters are estimated with gradient-based methods, without the support of an exact appearance model of the motion-blurred object over the background. In [6], the curved trajectory of a moving ball is reconstructed from a single, long-exposure image: in this case, an accurate analysis of the blurred ball streak is necessary to find its contours with sufficient precision for the subsequent 3D reconstruction step; this analysis is performed under some very restrictive assumptions, such as constant intensity of the ball image (*i.e.*, no shading), which are not needed when using our proposed technique; very similar considerations hold for [5], which analyzes a slightly-blurred ball and reconstructs its position and velocity. Recently, interesting related applications have also been proposed in [12], where rotational blur estimation is the basis for a visual gyroscope, and in [16], which exploit motion blur to obtain depth information from a single blurred image. Also, in [8] motion blur is used for joint reconstruction of geometry and photometry of scenes with multiple moving objects. In general, these applications take advantage of the additional information that a single motion blurred image incorporates about the scene structure, which can not be extracted from a single still image.

Many techniques ([4, 2], and recently [9]) have been proposed for estimating motion blur parameters, mainly at aimed at image "deblurring" (the first attempts date back to 1967 in [20]); spatially-variant motion blur has also been considered in many works, such as [11]. Our work is complementary to these techniques since we do not directly aim at interpreting the motion blur direction or extent; for example, our technique inherently handles complex, nonlinear trajectories or even object deformations; we operate at a lower level, separating the blurred object image from the background, and providing an "alpha map" which isolates the object motion, and a "color map" which is a blurred image of the object, separated from the background. We believe that in some settings, deblurring could actually take advantage of our resulting representation (see section 5).

We show in section 2.1 that part of our problem can be reduced to *alpha matting* (or *layer extraction*): a classic problem in computer vision which has been shown to be unconstrained in the general case; the subject has been extensively studied due to its immediate, obvious applications in many fields, and various solutions have been proposed: some ([21, 17]) require a specific background (blue screen matting), whereas others ([3, 7, 19, 22, 13, 1]), with minimal user assistance, handle unknown backgrounds (*natural image matting*) by means of sophisticated heuristics. None of these algorithms was designed for motion-blurred images – for example, all require that a part of the image has $\alpha = 1$, and many assume that mixed pixels are rare; nonetheless, some actually work acceptably in specific instances of this unforeseen scenario (see additional material), but quantitative use of the resulting alpha map is rarely an option.

2 Definitions and model

A motion blurred image is obtained when the scene projection on the image plane changes during the camera exposure period $e = [t_0, t_0 + \Delta t]$. The final image C is obtained as the integration of infinite sharp images, each exposed for an infinitesimal portion of e. An equivalent interpretation, which is more meaningful in this setting, is considering the motion blurred image as the temporal average of infinite sharp images I_t, each taken with the same exposure time Δt but representing the scene frozen at a different instant $t \in e$

(see figure 1). This technique is also implemented in many 3D rendering packages for accurate synthesis of motion blurred images.

If the camera is static and a single object is moving in the scene, the static background in the final image is sharp since its pixels are of constant intensity in each I_t; conversely, the image pixels which are affected by the moving object, possibly only in some of the I_t images, belong to the motion-blurred image of the object.

2.1 Motion blur and alpha compositing

For a pixel p, define $i(p) \subseteq e$ the time interval in e during which p belongs to the object image. We define $\alpha(p)$ as the fraction of e during which the object projects to p:

$$\alpha(p) = ||i(p)||/\Delta t. \tag{1}$$

Let $B(p)$ be the intensity of the background at p. Since C is the temporal average of all the I_t images, $C(p) = \alpha(p)o(p) + (1-\alpha(p))B(p)$, where $o(p)$ is the temporal average over $i(p)$ of the intensity of image pixel $C(p)$:

$$o(p) = \frac{1}{||i(p)||}\int_{t\in i(p)} I_t(p)\,dt. \tag{2}$$

To sum up, the intensity of a pixel p in the motion blurred image C can be interpreted as the convex linear combination of two factors: the "object" intensity $o(p)$, weighted $\alpha(p)$, and the background intensity. The resulting equation is the usual *over* Porter-Duff alpha compositing equation [18] for a pixel with transparency $\alpha(p)$ and intensity $o(p)$ over the background pixel $B(p)$.

$o(p)$ can be interpreted as the intensity that p would have in the motion blurred image over a black background, rescaled by a $\frac{1}{||i(p)||}$ factor. $o(p)$ is meaningless if p is not affected by the object image during e (i.e., if $\alpha(p) = 0$).

The considerations expressed so far can be directly applied to a color image, considering all channels separately, but noting that $\alpha(p)$ is constant along all the channels:

$$\begin{array}{ll} C_r(p) & = \alpha(p)o_r(p) + (1-\alpha(p))B_r(p) \\ C_g(p) & = \alpha(p)o_g(p) + (1-\alpha(p))B_g(p) \\ C_b(p) & = \alpha(p)o_b(p) + (1-\alpha(p))B_b(p). \end{array} \tag{3}$$

2.2 Assumptions on the object color

In order to provide an analytic reconstruction technique, we require that the object surface is monochromatic, meaning that, in the HSV color space, its hue and saturation are fixed – whereas different brightness values (shades) of the color are allowed. In the RGB color space, this translates to requiring that the object surface colors all lie on a single line passing through $(R,G,B) = (0,0,0)$. In practice, this includes many real-world objects; note that black surface parts of the object are always allowed.

Under this assumption and ignoring possible mixed lighting, specularity or transparency of the object surface, the colors of the object image satisfy the same condition even if shading is taken into account: they all still lie on a single line passing through $(R,G,B) = (0,0,0)$ – whereas their actual hue and saturation depend on the white balance and light color. We call this line in RGB space l.

In the motion-blurred image, $o_{rgb}(p)$ is constrained to lie on l as well, because it is a linear combination of colors belonging to l; if l is defined in polar coordinates:

$$\begin{aligned} o_r(p) &= k \cdot \sin\phi\cos\theta \\ o_g(p) &= k \cdot \cos\phi\sin\theta \\ o_b(p) &= k \cdot \cos\phi. \end{aligned} \tag{4}$$

3 Analysis technique

3.1 Recovering the object color

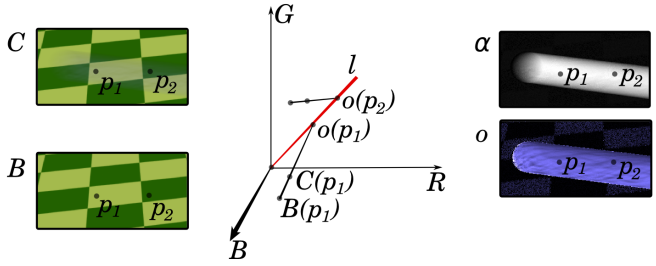


Figure 2: l (red line) passes through the origin and its orientation can be retrieved from the colors of C and B in two points. $o(p)$ is found as the intersection between l and the line passing through $C(p)$ and $B(p)$.

In a motion-blurred image, the object hue and saturation can be easily recovered if at least one pixel p exists such that $\alpha(p) = 1$; in other words, if p belongs to the image of the object during the whole exposure period, and is therefore not affected by the background color. In this case, the hue and saturation of $C(p)$ univocally define l in the RGB space.

Else, no pixel in the blurred image has a color which can be directly related to l. Therefore, we consider how the color of pixels belonging to the motion blurred images change between the background image B and the motion-blurred image C. Given two pixels p_1 and p_2 belonging to the object streak, we can write a system in 12 equations and 12 unknowns (k_1, k_2, ϕ, θ, $\alpha(p_1)$, $\alpha(p_2)$, the 6 $o_{rgb}(p_1)$ and $o_{rgb}(p_2)$ values), which has a single solution (see Appendix in supplementary material). A much simpler interpretation is given below.

Geometric interpretation in RGB space For each pixel p, from (4) we know that $o(p)$ lies on the line in RGB space connecting $C(p)$ to $B(p)$. However, due to the object non-uniform color and shading, $o(p)$ depends on the actual pixel. Therefore lines in RGB space defined by $C(p_1), B(p_1)$ and $C(p_2), B(p_2)$ will, in general, not intersect.

We know from (2) that $o(p_1)$ and $o(p_2)$ both lie on the same line in RGB space, which also passes through black ($(R,G,B) = (0,0,0)$). Therefore, we compute l as the only line passing through black and intersecting lines defined by $C(p_1)B(p_1)$ and $C(p_2)B(p_2)$; in other words, l can be found as the intersection of two planes, defined by RGB points $C(p_1), B(p_1), (0,0,0)$ and $C(p_2), B(p_2), (0,0,0)$. This requires that $B(p_1)$ and $B(p_2)$ are

linearly independent vectors in RGB space (*i.e.*, they do not represent the same color with different shading).

3.2 Recovering object exposure time and average color for each pixel

For a pixel in the blurred image, we can compute $\alpha(p)$ and $o(p)$ if l is known:

$$\begin{array}{ll} o_r(p) & = k \cdot \sin\phi\cos\theta \\ o_g(p) & = k \cdot \cos\phi\sin\theta \\ o_b(p) & = k \cdot \cos\phi \\ C_r(p) & = \alpha(p)o_r(p) + (1-\alpha(p))B_r(p) \\ C_g(p) & = \alpha(p)o_g(p) + (1-\alpha(p))B_g(p) \\ C_b(p) & = \alpha(p)o_b(p) + (1-\alpha(p))B_b(p). \end{array} \tag{5}$$

It's 6 equations for 5 unknowns ($\alpha(p)$, k and the 3 $o_{rgb}(p)$ values), so the system is overdetermined.

Geometric interpretation in RGB space From equation (3), $C(p)$ is a linear combination of $B(p)$ and $o(p)$: therefore, the line in RGB space connecting $B(p)$ and $C(p)$ intersects l in $o(p)$. This identifies $o(p)$. Once $o(p)$ is known, $\alpha(p)$ is found as $\overline{B(p)C(p)}/\overline{B(p)o(p)}$.

Pixels belonging to the background are unchanged between $B(p)$ and $C(p)$, therefore $\alpha = 0$ unless $B(p)$ belongs to l, which is a degenerate case. Contrarily, if a pixel p belongs to the object image during the entire exposure time, $C(p)$ belongs to l, so $\alpha = 1$.

3.3 Extensions and limits

Computing $o(p)$ and $\alpha(p)$ using (5) is an overconstrained problem. This suggests that the same theoretical framework could be adapted to handle more general problems and/or relax the assumptions.

- The object monochromaticity assumption can be relaxed: the problem can be still solved if the object surface is bichromatic – meaning that it is composed of two different colors, their shades towards black, and their (convex) linear combinations. Therefore, the surface colors of the object lie on a plane in RGB space. Since that plane passes through black, also the possible colors of the shaded object in a sharp image belong to a plane π (albeit possibly different, if light is not white). In the blurred image, $o(p)$ colors, obtained as linear combination of colors belonging to π, still lie on π.

- The nonspecular shading assumption can be relaxed: specular highlights appearing on a monochromatic object surface would cause the space of possible $o(p)$ values to extend on a plane, similarly to the previous case.

- The uniform lighting assumption can be relaxed: if only two different colors of light sources are present in the scene, all $o(p)$ colors still lie on a plane.

In practice, any of these three ways to exploit the additional degree of freedom make the reconstruction procedure for α an exactly determined problem, instead of overdetermined. Algebraically, one of the equations constraining the $o(p)$ position is removed; in the geometric interpretation, $o(p)$ is found as the intersection between a line (passing through $C(p)$ and $B(p)$) and a plane (π) – instead of the intersection between two lines. The degenerate case is when $B(p)$ belongs to π.

However, unless some additional assumptions or approximations are introduced, there is no procedure for automatically determining π analogous to the one introduced in section 3.1 for finding l. In practical applications, π can be computed in a different, sharp image of the object, by directly picking two colors linearly independent in RGB; they could be also picked directly in the blurred image, if some pixels exist for which $\alpha(p) = 1$. Using more than two pixels would also allow a more robust estimation of π by least-squares or RANSAC.

Unfortunately, the additional degree of freedom can not be exploited to handle variations in background lighting, such as shadows. In fact, this can be interpreted geometrically as constraining that $B(p)$ lies on a line in RGB passing through $(0,0,0)$. This line lies on the same plane defined by l and $C(p)$; this explains why the resulting system of equations is not determined. However, if the background color $B(p)$ is allowed to change along a line which does not pass through $(0,0,0)$, $o(p)$ and α can still be computed. This has a practical application for pixels near a sharp discontinuity between two different background colors: these are often problematic because of several factors, including image compression and minimal camera rotations after the background has been sampled. By only constraining $B(p)$ to lie on a line passing through both background colors, this problem can be avoided.

So far, we assumed that the object in each of the I_t images is perfectly sharp, and we ignored the presence of "mixed pixels", originating from imperfect focus, and anyway always present along the object contours – a well known problem in matting research. However, our procedure inherently handles these cases, and works with defocused, "fuzzy" or even semitransparent objects[1].

4 Experimental results and application examples

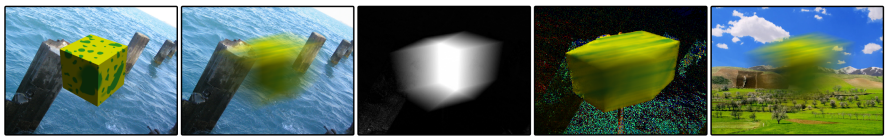


Figure 3: Synthetic example. From left to right: still bichromatic object, motion blurred image, reconstructed alpha map, reconstructed color map. Object composited over a different background using computed alpha and color map.

The technique has been evaluated both with synthetic images and with real images.

In figure 3, a bichromatic motion blurred cube is analyzed. Pixels in the central part of the streak belong to the object image during the whole exposure time, and are used

[1] Object transparency is handled under rather restrictive assumptions such as absence of refractive effects

to determine the plane of the possible object colors. The absence of any error in the reconstructed alpha and color maps confirms the theoretical validity of our approach. Once the alpha and color maps are known, the blurred object can be composited over a different background as shown; other interesting applications include artificial opacity or color manipulation – *e.g.* for visualizing long trajectories of balls in long-exposure photos.

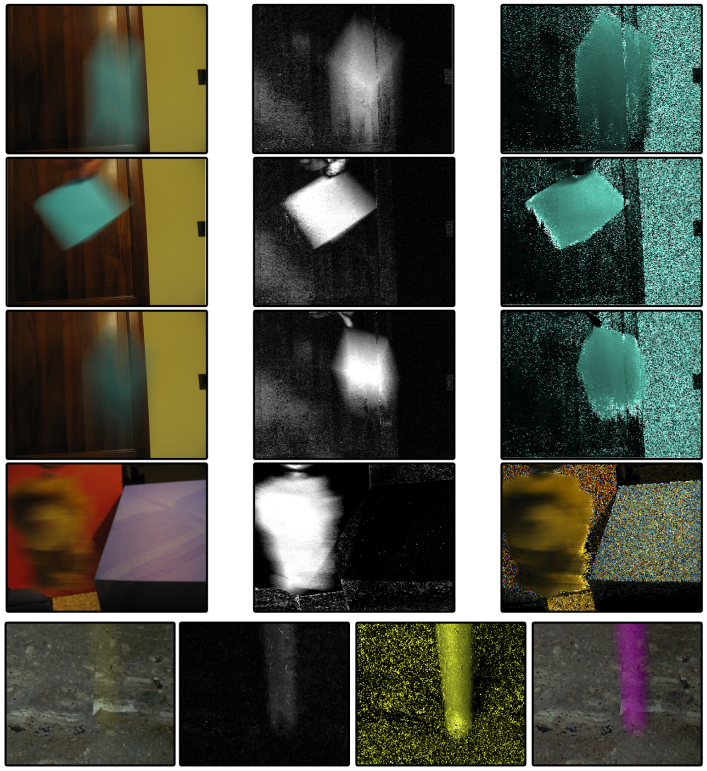


Figure 4: *First four lines*. From left to right: original image, alpha map, color map. Last object is bichromatic teddy bear, others are monochromatic small sheets of colored paper. Note that the color map is meaningless outside the object blurred image. *Last line*: part of the trajectory of a moving table tennis ball; from left to right: original image, alpha map, color map, enhanced image obtained by compositing a linearly-transformed color map over the background, using a multiplied alpha map for enhancing visibility (note preserved shading). See additional material for other examples and applications.

Figure 4 shows examples with real images. The reconstruction of a monochromatic object's color works reliably and robustly: in tests with table tennis balls and small colored paper sheets we were able to retrieve the color of the object image from a long streak over a multicolored background within an Euclidean distance of 3 units in the RGB color cube (each channel having $0 \div 255$ range) in all tests. Reconstructed alpha and color maps are heavily affected by image noise, which is expected since we work one pixel at a time

and do not yet perform any filtering. Also, discrepancies between the sampled and actual background introduce severe artifacts, which can be reduced, at least in part, with the technique described in section 3.3.

The theoretical background we presented builds upon the linearity of color channels, which is not normally preserved in images directly coming from digital cameras, since nonlinear gamma curves are applied automatically by the camera firmware unless the image is shot in RAW format. More generally, doing quantitative processing on the image colors requires that the camera response function is known. The estimation of the camera response function is the subject of several studies such as [10], which provide various convenient solutions which allow to linearize the intensity values of the color channels.

Often the reconstructed object color $o(p)$ exceeds the dynamic range of the image. This is fine, and means that, if the ball image projected to that pixel for the whole exposure time, it would have been overexposed; in this regard, note that accurate rendering of motion blur is an important application of High Dynamic Range image synthesis.

5 Conclusions and Future Developments

We formally defined the problem of reconstructing the motion blur of a moving object over a known background, describing how it relates to alpha matting, a deeply explored topic in computer vision.

If the object is monochromatic, we can retrieve its color even if it only appears as a semitransparent smear. If the object is monochromatic or bichromatic and its color is known, we can compute the fraction of the exposure time during which the object projection overlapped each pixel; also, we can determine how the image would appear if the background was black.

This allows, for example, to composite the blurred object over any background, artificially change its opacity or color or study patterns left by differently pigmented parts. Other applications which could take advantage from these techniques are related to the reconstruction of the object's motion from a single blurred image, and point spread function estimation problems for image restoration and deblurring.

We are currently applying these techniques in practice, while developing algorithms to robustly exploit them in presence of noise and other nonidealities. We are also using these techniques for motion estimation and 3D reconstruction from a single motion blurred image, by exploiting the alpha map's interpretation as the fraction of exposure time during which the object image overlapped that pixel. This allows, among others, applications in temporal superresolution of apparent contours motion. We are also applying known blind deblurring techniques on the alpha and color maps separately, in order to precisely deblur objects on a fixed, sharp background.

References

[1] N. E. Apostoloff and A. W. Fitzgibbon. Bayesian video matting using learnt image priors. In *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[2] B. Bascle, Andrew Blake, and Andrew Zisserman. Motion deblurring and super-resolution from an image sequence. In *Proc. of European Conference on Computer Vision (ECCV)*, pages 573–582, 1996.

[3] A. Berman, P. Vlahos, and A. Dadourian. Comprehensive method for removing from an image the background surrounding a selected object. U.S. Patent 6,134,345, 2000.

[4] M. Bertero and P. Boccacci. *Introduction to Inverse Problems in Imaging*. Inst. of Physics Publishing, 1998.

[5] Giacomo Boracchi, Vincenzo Caglioti, and Alessandro Giusti. Ball position and motion reconstruction from blur in a single perspective image. In *Proc. of International Conference on Image Analysis and Processing (ICIAP)*, 2007.

[6] V. Caglioti and A. Giusti. Ball trajectory reconstruction from a single long-exposure perspective image. In *Proc. of Workshop on Computer Vision Based Analysis in Sport Environments (CVBASE)*, 2006.

[7] Y. Chuang, B. Curless, D. Salesin, and R. Szeliski. A bayesian approach to digital matting. In *Proc. of CVPR 2001*.

[8] Paolo Favaro and Stefano Soatto. A variational approach to scene reconstruction and image segmentation from motion-blur cues. In *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 01, pages 631–637, Los Alamitos, CA, USA, 2004. IEEE Computer Society.

[9] Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T. Roweis, and William T. Freeman. Removing camera shake from a single photograph. In *ACM SIGGRAPH 2006 Papers*, pages 787–794, New York, NY, USA, 2006. ACM Press.

[10] Michael D. Grossberg and Shree K. Nayar. Modeling the space of camera response functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1272–1282, 2004.

[11] S.K. Kang, Y.C. Choung, and J.K. Paik. Segmentation-based image restoration for multiple moving objects with different motions. In *Proc. of International Conference on Image Processing (ICIP)*, pages I:376–380, 1999.

[12] G. Klein and T. Drummond. A single-frame visual gyroscope. In *Proc. of British Machine Vision Conference (BMVC)*, 2005.

[13] Anat Levin, Dani Lischinski, and Yair Weiss. A closed form solution to natural image matting. In *Proc. of CVPR 2006*, pages 61–68, Washington, DC, USA, 2006. IEEE Computer Society.

[14] Huei-Yung Lin. Vehicle speed detection and identification from a single motion blurred image. In *Proc. of IEEE Workshop on Applications of Computer Vision*, pages 461–467, 2005.

[15] Huei-Yung Lin and Chia-Hong Chang. Automatic speed measurements of spherical objects using an off-the-shelf digital camera. In *Proc. of IEEE International Conference on Mechatronics*, pages 66–71, 2005.

[16] Huei-Yung Lin and Chia-Hong Chang. Depth recovery from motion blurred images. In *Proc. of International Conference on Pattern Recognition (ICPR)*, volume 1, pages 135–138, Los Alamitos, CA, USA, 2006. IEEE Computer Society.

[17] Y. Mishima. Soft edge chroma-key generation based upon hexoctahedral color space. U.S. Patent 5,355,174, 1993.

[18] T. Porter and T. Duff. Compositing digital images. *Computer Graphics*, 1984.

[19] M. Ruzon and C. Tomasi. Alpha estimation in natural images. In *Proc. of CVPR 2000*.

[20] D. Slepian. Restoration of photographs blurred by image motion. *Bell System Tech.*, 46(10):2353–2362, December 1967.

[21] Alvy Ray Smith and James F. Blinn. Blue screen matting. In *SIGGRAPH '96: Proc. of the 23rd annual conference on Computer graphics and interactive techniques*, pages 259–268, New York, NY, USA, 1996. ACM Press.

[22] Jian Sun, Jiaya Jia, Chi-Keung Tang, and Heung-Yeung Shum. Poisson matting. In *ACM SIGGRAPH 2004 Papers*, pages 315–321, New York, NY, USA, 2004. ACM Press.

Improved Face Model Fitting on Video Sequences [†]

Xiaoming Liu Frederick W. Wheeler Peter H. Tu

Visualization and Computer Vision Lab
General Electric Global Research Center
Niskayuna, NY 12309, USA
{liux,wheeler,tu}@research.ge.com

Abstract

Active Appearance Models (AAMs) represent the shape and appearance of an object via two low-dimensional subspaces, one for shape and one for appearance. AAMs for facial images are currently receiving considerable attention from the computer vision community. However, most existing work focuses on fitting AAMs to a single image. For many applications, effectively fitting an AAM to video sequences is of critical importance and challenging, especially considering the varying quality of real-world video content. This paper proposes a hybrid model to address this problem. Both a generic AAM and a subject-specific model are employed simultaneously in the proposed fitting scheme. Experimental results from outdoor surveillance video sequences demonstrate the improved image registration across video frames and faster fitting convergence.

1 Introduction

Model-based image registration/alignment is a fundamental topic in computer vision. *Active Appearance Models (AAMs)* have been one of the most popular models for image registration [4]. Face alignment using an AAM is receiving considerable attention from the computer vision community because it enables various capabilities such as facial feature detection, pose rectification, and gaze estimation. However, most existing work focuses on fitting the AAM to a single facial image. With the abundance of surveillance cameras and greater need for face recognition from video, methods to effectively fit an AAM to facial images in videos are of increasing importance. This paper addresses this problem and proposes a novel algorithm for it.

There are two basic components in face alignment using an AAM: *face modeling* and *model fitting*. Given a set of facial images, *face modeling* is the procedure of training the AAM, which is essentially two distinct linear subspaces modeling facial shape and

[†]This project was supported by awards #2005-IJ-CX-K060 and #2006-IJ-CX-K045 awarded by the National Institute of Justice, Office of Justice Programs, US Department of Justice. The opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the Department of Justice.

appearance respectively. *Model fitting* refers to estimating the parameters of the resulting AAM on faces in an image or video frames by minimizing the distance measured between the image and the AAM.

In the context of fitting an AAM to video sequences, conventional methods directly fit the AAM to each frame by using the fitting results, i.e., the shape and appearance parameters, of the previous frame as the initialization of the current frame. However, as shown in the previous work [6], fitting to faces of an unseen subject can be hard due to the mismatch between the appearance of the facial images used for training the AAM and that of the video sequences, especially when the video sequences are captured in the outdoor environment. Also, the conventional method only registers each frame with respect to the AAM, without enforcing the frame-to-frame registration across video sequences, which is necessary for many practical applications, such as multi-frame super-resolution [13].

To address this problem, we propose a novel approach to continuously fit the AAM to video sequences. The proposed algorithm is an extension of the state-of-the-art image alignment algorithm – the Simultaneous Inverse Compositional (SIC) method [1], which minimizes the distance of the warped image observation and the generic AAM model during the fitting. We call our proposed approach as "*SIC fOr Video (SICOV)*" algorithm, which not only minimizes the above distance measure, but also the distance between the warped image and a model obtained from the warped images of previous video frames. Experimental results show that the SICOV algorithm improves both the fitting accuracy across frames and the fitting speed.

Many approaches have been proposed for modeling faces with AAMs [4, 1]. Baker and Matthews [1] proposed the Inverse Compositional (IC) method and SIC method that greatly improves the fitting speed and performance. However, little work has been done in fitting AAMs to facial video sequences in particular. Koterba *et al.* [7] proposed to use a 3D face model as a constraint in fitting multiple video frames. Matthews *et al.* [11] also updated the generic AAM using the warped image observation, such that a subject-specific model can be obtained during the fitting process. Comparing to their approach, we will show that treating the previous frame information as an additional constraint can improve the fitting speed, not to mention saving the extra time needed to update the bulky eigenspace of the appearance model in an AAM. Bosch *et al.* [2] proposed an Active Appearance Motion Model that captures the motion pattern in video sequences by taking the concatenation of the landmarks from multiple frames as training samples. This approach takes advantage of the periodic motion pattern in medical image sequences. In contrast, our approach does not make assumption on the object's motion.

This paper is organized as follows. Section 2 introduces the conventional methods for training the AAM and model fitting. Sections 3 and 4 present the proposed SICOV algorithm and its detailed derivation. Section 5 provides experimental results, and conclusions are given in Section 6.

2 Active Appearance Models and Model Fitting

The shape model and appearance model part of an AAM are trained with a representative set of facial images. The distribution of facial landmarks are modeled as a Gaussian distribution, which is regarded as the shape model. The procedure for training a shape model is as follows. Given a face database, each facial image is manually labeled with

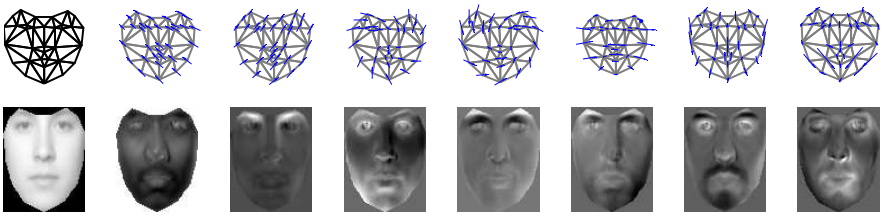


Figure 1: The mean and first 7 basis vectors of the shape model (top) and the appearance model (bottom) trained from the ND1 database. The shape basis vectors are shown as arrows at the corresponding mean shape landmark locations.

a set of 2D landmarks, $[x_i, y_i]$ $i = 1, 2, ..., v$. The collection of landmarks of one image is treated as one observation from the random process defined by the shape model, $\mathbf{s} = [x_1, y_1, x_2, y_2, ..., x_v, y_v]^T$. Eigen-analysis is applied to the observation set and the resulting linear shape model represents a shape as,

$$\mathbf{s}(\mathbf{P}) = \mathbf{s}_0 + \sum_{i=1}^{n} p_i \mathbf{s}_i, \tag{1}$$

where $\mathbf{s}_0$ is the mean shape, $\mathbf{s}_i$ is the i^{th} shape basis, and $\mathbf{p} = [p_1, p_2, ..., p_n]$ are the shape parameters. By design, the first four shape basis vectors represent global rotation and translation. Together with other basis vectors, a mapping function from the model coordinate system to the coordinates in the image observation is defined as $\mathbf{W}(\mathbf{x}; \mathbf{p})$, where $\mathbf{x}$ is a pixel coordinate defined by the mean shape $\mathbf{s}_0$.

After the shape model is trained, each facial image is warped into the mean shape using a piecewise affine transformation. These shape-normalized appearances from all training images are fed into an eigen-analysis and the resulting model represents an appearance as,

$$A(\mathbf{x}; \lambda) = T(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i A_i(\mathbf{x}), \tag{2}$$

where T is the mean appearance, A_i is the i^{th} appearance basis, and $\lambda = [\lambda_1, \lambda_2, ..., \lambda_m]$ are the appearance parameters. Figure 1 shows an AAM trained using 534 images of 200 subjects from the ND1 3D face database [3].

An AAM can synthesize facial images with arbitrary shape and appearance within the range expressed by the training population. Thus, the AAM can be used to *explain* a facial image by finding the optimal shape and appearance parameters such that the synthesized image is as similar to the image observation as possible. This leads to the cost function used for model fitting [5],

$$J(\mathbf{p}, \lambda) = \sum_{\mathbf{x} \in \mathbf{s}_0} [I(\mathbf{W}(\mathbf{x}; \mathbf{p})) - A(\mathbf{x}; \lambda)]^2, \tag{3}$$

which is the mean-square-error (MSE) between the image warped from the observation $I(\mathbf{W}(\mathbf{x}; \mathbf{p}))$ and the synthesized appearance model instance $A(\mathbf{x}; \lambda)$.

Traditionally this minimization problem is solved by iterative gradient-descent methods which estimate $\Delta\mathbf{p}$, $\Delta\lambda$ and add them to $\mathbf{p}$, λ. Baker and Matthews [1] proposed the

compositional method to generate the new shape parameter based on $\Delta\mathbf{p}$ in their IC and SIC method. The key idea of IC and SIC is that the role of the appearance template and the input image is switched when computing $\Delta\mathbf{p}$. This enables the time-consuming steps of parameter estimation to be pre-computed and performed outside of the iteration loop. We will borrow this key idea in deriving the solution of our SICOV algorithm.

3 The SICOV algorithm

Given a generic AAM and a video frame I_t at time t, SICOV uses the following cost function to perform the face model fitting:

$$J_t(\mathbf{p},\lambda) = \sum_{\mathbf{x}\in\mathbf{s}_0} [T(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i A_i(\mathbf{x}) - I_t(\mathbf{W}(\mathbf{x};\mathbf{p}))]^2 + k \sum_{\mathbf{x}\in\mathbf{s}_0} [M_t(\mathbf{x}) - I_t(\mathbf{W}(\mathbf{x};\mathbf{p}))]^2, \quad (4)$$

which is composed of two terms weighted by a constant k. The first one is the same as Eq. (3), i.e., the MSE between the warped image and the synthesized appearance model instance. The second one is the MSE between the current warped image $I_t(\mathbf{W}(\mathbf{x};\mathbf{p}))$ and the appearance information of the current subject from previous frames, $M_t(\mathbf{x})$.

There are different options in defining $M_t(\mathbf{x})$. Firstly, it can be the warped image of the video frame at time $t-1$:

$$M_t(\mathbf{x}) = I_{t-1}(\mathbf{W}(\mathbf{x};\mathbf{p}_{t-1})). \quad (5)$$

Secondly, the warped images of L previous video frames averaged by a decaying factor can also represent $M_t(\mathbf{x})$:

$$M_t(\mathbf{x}) = \frac{1-r}{r(1-r^L)} \sum_{l=1}^{L} r^l I_{t-l}(\mathbf{W}(\mathbf{x};\mathbf{p}_{t-l})), \quad (6)$$

where r is a decaying factor between 0 and 1. In practice, when fitting the video frame at time t, both definitions of $M_t(\mathbf{x})$ are known and can be computed efficiently from the previous fitting results. Of course, other definitions of $M_t(\mathbf{x})$ are also possible, for example, the average of L previous warped images without decaying, and a dynamic eigenspace model of the previous warped images [9]. In the latter case, an efficient eigenspace updating method can be used to sequentially add the most recent warped image into the model [8], and additional appearance parameters of this eigenspace model should be incorporated into the the second term of Eq. (4).

These two terms in Eq. (4) can be treated as the distance between the current image observation and the generic face model and the subject-specific model respectively, which is obtained in an on-line fashion from image observation at the previous time instances. Thus in the fitting of each frame, both distance measures are served as constraints to guide the fitting process.

There are clear benefits from using these two models during the face model fitting. First of all, in practical applications there is always mismatch between the imaging environment of the images used for training face models and the images to be fit, as well as the presence of the specific appearance information of the subject being fit that is not modeled by the generic face models. Thus the distance-to-subject-specific-model is employed to bridge such a gap. Secondly, if we only use the subject-specific model, the alignment error would propagate over time. The generic model is well suited for preventing the error propagation and correcting the drifting.

4 Derivation of SICOV algorithm

Using an approach similar to the IC and SIC algorithms [1], the proposed SICOV algorithm iteratively minimizes:

$$\sum_{\mathbf{x}}\left[T(\mathbf{W}(\mathbf{x};\triangle\mathbf{p}))+\sum_{i=1}^{m}(\lambda_i+\triangle\lambda_i)A_i(\mathbf{W}(\mathbf{x};\triangle\mathbf{p}))-I_t(\mathbf{W}(\mathbf{x};\mathbf{p}))\right]^2 +k\sum_{\mathbf{x}}\left[M_t(\mathbf{W}(\mathbf{x};\triangle\mathbf{p}))-I_t(\mathbf{W}(\mathbf{x};\mathbf{p}))\right]^2 \tag{7}$$

with respect to $\triangle\mathbf{p}$ and $\triangle\lambda=(\triangle\lambda_1,...,\triangle\lambda_m)^{\mathrm{T}}$ simultaneously, and then updates the warp $\mathbf{W}(\mathbf{x};\mathbf{p})\leftarrow\mathbf{W}(\mathbf{x};\mathbf{p})\circ\mathbf{W}(\mathbf{x};\triangle\mathbf{p})^{-1}$ and the appearance parameter $\lambda\leftarrow\lambda+\triangle\lambda$.

In order to solve for $\triangle\mathbf{p}$ and $\triangle\lambda$, the non-linear expression in Eq. (7) is linearized by performing a first order Taylor series expansion on $T(\mathbf{W}(\mathbf{x};\triangle\mathbf{p}))$, $A_i(\mathbf{W}(\mathbf{x};\triangle\mathbf{p}))$, and $M_t(\mathbf{W}(\mathbf{x};\triangle\mathbf{p}))$, and assuming that $\mathbf{W}(\mathbf{x};\mathbf{0})$ is the identity warp. This gives:

$$\sum_{\mathbf{x}}\left[T(\mathbf{x})+\nabla T\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\triangle\mathbf{p}+\sum_{i=1}^{m}(\lambda_i+\triangle\lambda_i)(A_i(\mathbf{x})+\nabla A_i\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\triangle\mathbf{p})-I_t(\mathbf{W}(\mathbf{x};\mathbf{p}))\right]^2 +k\sum_{\mathbf{x}}\left[M_t(\mathbf{x})+\nabla M_t\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\triangle\mathbf{p}-I_t(\mathbf{W}(\mathbf{x};\mathbf{p}))\right]^2. \tag{8}$$

The first term in the above equation can be simplified as follows by neglecting the second order terms:

$$\sum_{\mathbf{x}}\left[T(\mathbf{x})+\sum_{i=1}^{m}\lambda_iA_i(\mathbf{x})-I_t(\mathbf{W}(\mathbf{x};\mathbf{p}))+(\nabla T+\sum_{i=1}^{m}\lambda_i\nabla A_i)\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\triangle\mathbf{p}+\sum_{i=1}^{m}A_i(\mathbf{x})\triangle\lambda_i\right]^2. \tag{9}$$

To simplify the notation, firstly we denote $\mathbf{q}=(\mathbf{p}^{\mathrm{T}}\lambda^{\mathrm{T}})^{\mathrm{T}}$ and similarly $\triangle\mathbf{q}=(\triangle\mathbf{p}^{\mathrm{T}}\triangle\lambda^{\mathrm{T}})^{\mathrm{T}}$. Thus $\mathbf{q}$ is a $n+m$ dimensional vector including both the shape parameters $\mathbf{p}$ and the appearance parameters λ. Secondly, we denote $n+m$ dimensional steepest-decent images:

$$\mathbf{SD}(\mathbf{x})=\left[(\nabla T+\sum_{i=1}^{m}\lambda_i\nabla A_i+k\nabla M_t)\frac{\partial\mathbf{W}}{\partial p_1},...,(\nabla T+\sum_{i=1}^{m}\lambda_i\nabla A_i+k\nabla M_t)\frac{\partial\mathbf{W}}{\partial p_n},A_1(\mathbf{x}),...,A_m(\mathbf{x})\right]. \tag{10}$$

Thirdly, we denote the error image:

$$E(\mathbf{x})=T(\mathbf{x})+\sum_{i=1}^{m}\lambda_iA_i(\mathbf{x})-I_t(\mathbf{W}(\mathbf{x};\mathbf{p}))+k(M_t(\mathbf{x})-I_t(\mathbf{W}(\mathbf{x};\mathbf{p}))). \tag{11}$$

Equation (8) is simplified to:

$$\sum_{\mathbf{x}}[E(\mathbf{x})+\mathbf{SD}(\mathbf{x})\triangle\mathbf{q}]^2. \tag{12}$$

The partial derivative of Eq. (12) with respect to $\triangle\mathbf{q}$ is:

$$2\sum_{\mathbf{x}}\mathbf{SD}^{\mathrm{T}}(\mathbf{x})[E(\mathbf{x})+\mathbf{SD}(\mathbf{x})\triangle\mathbf{q}]. \tag{13}$$

Pre-compute:

(3) Evaluate the gradients ∇T, ∇M_t, and ∇A_i for $i = 1, 2, ..., m$

(4) Evaluate the Jacobian $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ at $(\mathbf{x}; \mathbf{0})$

Iterate:

(1) Warp I with $\mathbf{W}(\mathbf{x}; \mathbf{p})$ to compute $I(\mathbf{W}(\mathbf{x}; \mathbf{p}))$

(2) Compute the error image $E(\mathbf{x})$ using Eq. (11)

(5) Compute the steepest decent image $\mathbf{SD}(\mathbf{x})$ using Eq. (10)

(6) Compute the Hessian matrix $\mathbf{H}$ using Eq. (15) and invert the matrix

(7) Compute $\sum_{\mathbf{x}} \mathbf{SD}^{\mathrm{T}}(\mathbf{x}) E(\mathbf{x})$

(8) Compute $\triangle \mathbf{q}$ using Eq. (14)

(9) Update $\mathbf{W}(\mathbf{x}; \mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x}; \mathbf{p}) \circ \mathbf{W}(\mathbf{x}; \triangle \mathbf{p})^{-1}$ and $\lambda \leftarrow \lambda + \triangle \lambda$

until $||\triangle \mathbf{p}|| \leq \varepsilon$

Figure 2: Summary of the SICOV algorithm.

The closed form solution of Eq. (7) is obtained by setting Eq. (13) to equal zero:

$$\triangle \mathbf{q} = -\mathbf{H}^{-1} \sum_{\mathbf{x}} \mathbf{SD}^{\mathrm{T}}(\mathbf{x}) E(\mathbf{x}), \tag{14}$$

where $\mathbf{H}^{-1}$ is the inverse of the Hessian matrix:

$$\mathbf{H} = \sum_{\mathbf{x}} \mathbf{SD}^{\mathrm{T}}(\mathbf{x}) \mathbf{SD}(\mathbf{x}). \tag{15}$$

The algorithm is summarized in Figure 2. The computation cost of the SICOV algorithm is summarized in Table 1. It can be seen that although the additional constraint results in slight more computation in Step (2) and Step (5), the computation cost per iteration of SICOV is almost the same as that of the SIC algorithm [1].

Pre-computation	Step 3	$O(mN)$	$O((n+m)N)$
	Step 4	$O(nN)$	
Per Iteration	Step 1	$O(nN)$	$O((n+m)^2 N + (n+m)^3)$
	Step 2	$O(mN)$	
	Step 5	$O((n+m)N)$	
	Step 6	$O((n+m)^2 N + (n+m)^3)$	
	Step 7	$O((n+m)N)$	
	Step 8	$O((n+m)^2)$	
	Step 9	$O(n^2+m)$	

Table 1: The computation cost of the SICOV algorithm. The right column indicates the total cost for the pre-computation and each iteration.

5 Experiments

To evaluate our algorithm, we collect a set of 400 images from two public available databases, the ND1 database [3], which contains 953 facial images with mostly frontal

Figure 3: Examples of the face dataset: ND1 database (left) and FERET database (right).

views from 273 subjects, and the FERET database [12], which contains a large number of subjects with various poses and expressions. Figure 3 shows sample images from these two databases. In our experiment, we use a 200-image subset from the ND1 database and a 200-image subset from the FERET database. Each one of the 400 images comes from different subjects. This 400-image set is used to train a generic AAM. Iterative model enhancement [10] is used in the training stage and results in a more compact model than the conventional approach. The resulting AAM has 10 shape bases, 52 appearance bases, and the width of the mean shape is 62 pixels.

A number of outdoor test surveillance video sequences, whose subjects are not included in the training dataset, are captured at 30 frames per second (FPS). For comparison purpose, we have implemented both the SIC and SICOV algorithms in Matlab^{TM}. By manually placing the mean shape on the first video frame, SICOV and SIC algorithms are used to fit the above generic AAM to these test videos respectively. The only parameter for the SICOV algorithm, k, is set to $k = 1$ throughout the experiments. Ideally k should be set according on the *correctness* of the individual model $M_t(\mathbf{x})$. We use Eq. (5) as the definition of $M_t(\mathbf{x})$. The first video sequence contains 980 frames. The proposed SICOV algorithm successfully fits the face over the whole video sequence while the SIC algorithm loses the fitting starting from frame 780 due to large pose change. In the case where there is no manual label for each frame of the test video sequences, visual inspection of the fitting results is one way of evaluating the performance. Figure 4 shows the comparison between two methods on 6 frames in this video. A visually more accurately fitted mesh is observed when using the SICOV algorithm.

Other than visual inspection, an alternative way to evaluate the fitting performance is to quantitatively compute the registration consistency across frames, which is represented by the MSE of the warped image observations between consecutive frames. As shown in Figure 5, SICOV provides on average lower MSE for the entire sequence, especially when SIC has high MSE at certain frames due to the changing facial appearance. Hence this shows superior frame-to-frame registration using the SICOV algorithm. On one hand, this is a favorable property for many applications that requires accurate registration across time, such as super resolution from video sequences. On the other hand, this is also an expected result since the frame-to-frame registration measure is part of the SICOV's objective function.

Our proposed method can improve not only the fitting robustness and accuracy, but also the fitting speed. Figure 6 shows the number of iterations for fitting each frame using the SIC and SICOV algorithm. The lower curve of SICOV indicates that SICOV can converge much faster than SIC. This improvement is expected because the additional

Figure 4: Comparison of the fitted mesh using the SICOV algorithm (dashed line) and the conventional SIC algorithm (solid line) on 6 frames (frame 1, 40, 87, 287, 734 and 767).

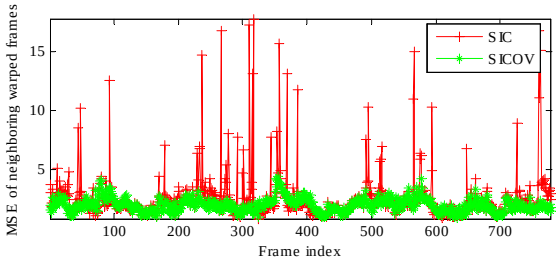


Figure 5: The MSE of neighboring warped frames of a video sequence. Constant lower MSE indicates the improved frame-to-frame registration using the SICOV algorithm.

constraint in SICOV helps the minimization procedure. Given the fact that the computation cost per iteration in the fitting is almost the same as SIC, the average time for fitting one frame using SICOV is much lower because less iterations are needed for fitting to converge. Based on a Matlab^{TM} implementation running on a conventional 2.13 GHz Pentium^{TM}4 computer, on average SICOV takes 0.1254 sec. to fit one frame compared to 0.2526 sec. by SIC. We have also implemented the SICOV using C++ and resulting facial fitting system can run 25+ FPS on a conventional PC for unseen subjects.

Figure 7 shows the fitting results on another 970-frame-long video sequence, where a Pan-Tilt-Zoom (PTZ) camera is pointing at three subjects and continuously zooming out. This is to mimic the scenario where in surveillance applications the subjects can have various distance to a camera and the face image can be of low resolution. How to effectively fit a face model onto this type of challenging real-world video sequence receives relatively little attention in the vision community. The proposed SICOV algorithm successfully fits the entire video sequence, even when zooming happens and large scale change appears in consecutive frames. Note that the smallest face size in this video sequence only has the face width of 15 pixels. However, when applying the conventional SIC algorithm, the fitting diverges starting at frame 34 when the first zooming happens.

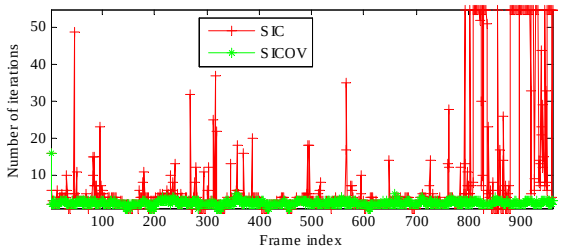


Figure 6: The number of iterations in fitting each frame of a video sequence. Constant lower number of iterations is observed from the proposed SICOV algorithm.

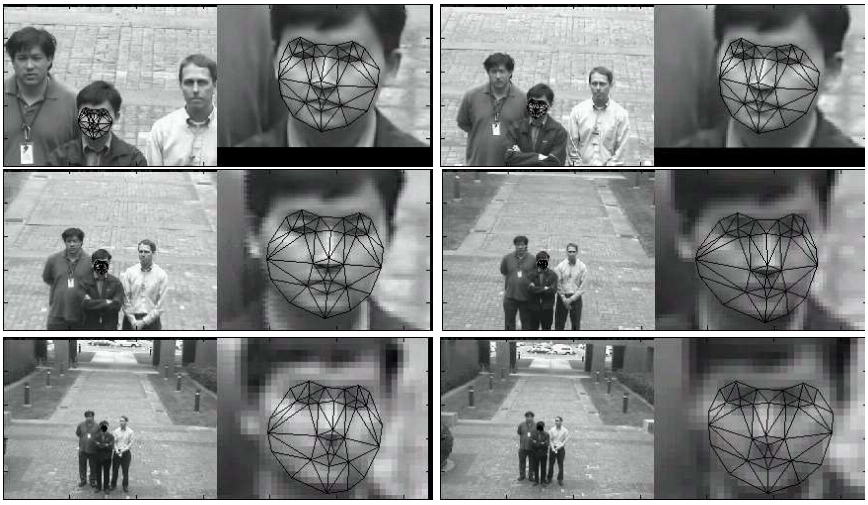


Figure 7: Fitting results with zoom in facial area using SICOV. Reliable fitting is observed in dealing with zooming and low resolution, even for the facial area of 15 pixels wide (lower right).

6 Conclusions

This paper studies methods to effectively fit an AAM to facial video sequences by using a hybrid model. Both a generic AAM and a subject-specific model are employed simultaneously in the proposed fitting scheme. Borrowing the idea of the SIC algorithm, we also introduce the efficient implementation of the proposed algorithm. Experimental results from outdoor surveillance video sequences demonstrate the improved fitting robustness, accuracy and speed. Future directions of this work can be experimenting with other definitions of the subject-specific model, such as Eq. (6), and as well as investigating the option of dynamically determining the weighting factor k based on the observed video frame.

References

[1] S. Baker and I. Matthews. Lucas-Kanade 20 years on: A unifying framework. *Int. J. Computer Vision*, 56(3):221–255, March 2004.

[2] Johan G. Bosch, Steven C. Mitchell, Boudewijn P. F. Lelieveldt, Francisca Nijland, Otto Kamp, Milan Sonka, and Johan H. C. Reiber. Automatic segmentation of echocardiographic sequences by active appearance motion models. *IEEE Trans. Medical Imaging*, 21(11):1374–1383, 2002.

[3] K. Chang, K. Bowyer, and P. Flynn. Face recognition using 2D and 3D facial data. In *Proc. ACM Workshop on Multimodal User Authentication*, pages 25–32, December 2003.

[4] T. Cootes, D. Cooper, C. Tylor, and J. Graham. A trainable method of parametric shape description. In *Proc. 2nd British Machine Vision Conference, Glasgow, UK*, pages 54–61. Springer, September 1991.

[5] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(6):681–685, June 2001.

[6] R. Gross, I. Matthews, and S. Baker. Generic vs. person specific active appearance models. *Image and Vision Computing*, 23(11):1080–1093, November 2005.

[7] Seth C. Koterba, Simon Baker, Iain Matthews, Changbo Hu, Jing Xiao, Jeffrey Cohn, and Takeo Kanade. Multi-view AAM fitting and camera calibration. In *Proc. 10th Int. Conf. on Computer Vision, Beijing, China*, volume 1, pages 511–518, October 2005.

[8] A. Levey and M. Lindenbaum. Sequential Karhunen-Loeve basis extraction and its application to images. *IEEE Trans. Image Processing*, 9(8):1371–1374, 2000.

[9] Xiaoming Liu, Tsuhan Chen, and Susan M. Thornton. Eigenspace updating for non-stationary process and its application to face recognition. *Pattern Recognition*, 36(9):1945–1959, 2003.

[10] Xiaoming Liu, Peter Tu, and Frederick Wheeler. Face model fitting on low resolution images. In *Proc. 17th British Machine Vision Conference, Edinburgh, UK*, volume 3, pages 1079–1088, 2006.

[11] Iain Matthews, Takahiro Ishikawa, and Simon Baker. The template update problem. In *Proc. 14th British Machine Vision Conference, Norwich, UK*, September 2003.

[12] P. J. Phillips, H. Moon, P. J. Rauss, and S. Rizvi. The FERET evaluation methodology for face recognition algorithms. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, October 2000.

[13] Frederick W. Wheeler, Xiaoming Liu, Peter H. Tu, and Ralph Hoctor. Multi-frame image restoration for face recognition. In *Proc. IEEE Signal Processing Society Workshop on Signal Processing Applications for Public Security and Forensics (SAFE 2007), Washington, DC*, 2007.

Fitting Surface of Free Form Objects using Optimized NURBS Patches Network with Evolutionary Strategies $(\mu + \lambda) - ES$

John William Branch
Escuela de Sistemas
Universidad Nacional de Colombia - Sede Medellín, Colombia
jwbranch@unalmed.edu.co

Flavio Prieto
Departamento de Eléctrica, Electrónica y Computación
Universidad Nacional de Colombia - Sede Manizales, Colombia
faprietoo@unal.edu.co

Pierre Boulanger
Department of Computing Science
University of Alberta, Canada
pierreb@cs.ualberta.ca

Abstract

We propose an algorithm to produce a 3-D CAD model from a set of range data, based on non-uniform rational B-splines (NURBS) surface fitting technique. Our goal is to construct continuous geometric models, assuming that the topology of surface is unknown. In our approach, the triangulated surface is partitioned in quadrilateral patches, using Morse theory. The quadrilateral obtained mesh is regularized by means of the use of geodesic curves and B-splines to obtain a new adequate grid on which to draw NURBS surfaces. Such NURBS surfaces are optimized by means of evolutionary strategies. Further, the patches are smoothly joined guaranteeing continuity C^1.

1 Introduction

Computer-aided geometric design and computer-aided manufacturing systems are used in numerous industries to design and create physical objects from digital models. Typically, the process consist of constructing complex objects by a combination of simple geometrical primitives. Many of these primitives are combined by boolean operations or by specifying a boundary representation where the topology and the geometry of the object are well known. However the reverse problem, which is of inferring a geometric model from an existing physical object digitized by a 3-D sensor, is a much harder problem as it is ill-posed. This paper addresses the problem of recovering 3D shape by using NURBS surfaces defined topologically as a network of quadrilaterals curves over the surface. The specification of the problem to be solved can be stated as follows:

"Given a set of sample points X assumed to lie on or near an unknown surface U, create a surface model S approximating U" [7].

In the general surface reconstruction problem, we consider that the points X are noisy. No structure or other information is assumed. The surface U -assumed to be a manifold- may have arbitrary topology, including boundaries, they contain sharp features such as creases and corners. Since the points X are noisy samples, we do not attempt to interpolate them, but instead find an approximating surface. Of course, a surface reconstruction procedure cannot guarantee recovering U exactly, since it is only given information

about U through a finite set of noisy sample points. The reconstructed surface S should have the same topological type as U and be close to U.

This paper is organized as following: Section 2, describes the Morse theory for triangular meshes. Section 3, a review of the pertinent literature in 3-D reconstruction. Section 4 describes the method for the adjustment of surfaces by means of optimized NURBS patches. Section 5 describes the experimental results of the proposed algorithm, and finally, in Section 6 a conclusion is presented.

2 Morse Theory for Triangular Meshes

The application of the Morse theory for triangular meshes implies to discretize Morse analysis. The Laplacian equation is used to find a Morse function which describes the topology represented on the triangular mesh. In this sense, additional points of the feature of the surface might exist, which produce a basis domain which adequately represents the geometry of the topology itself and the original mesh. The mesh can also be grouped into improved patches. In this work, Morse theory is applied by representing the saddle points and its borders by a Morse function which can then be used to determine a number of critical points.

This approximation function is based on a discrete version of the Laplacian, to find the harmonic functions. In many ways, Morse theory relates the topology of a surface S with its differential structure specified by the critical points of a Morse function $h : S \rightarrow \mathbb{R}$ [17] and is related to the mesh spectral analysis.

The spectral analysis of the mesh is performed by initially calculating the Laplacian. The discrete Laplacian operator on piecewise linear functions over triangulated manifolds is given by:

$$\Delta f_i = \sum_{j \in N_i} W_{ij}(f_j - f_i) \tag{1}$$

where N_i is the set of vertices adjacent to vertex i and W_{ij} is a scalar weight assigned to the directed edge (i, j). For graphs free of any geometry embedding, it is customary to use the combinatorial weights $W_{ij} = 1/\deg(i)$ in defining the operator. However, for 2-manifold mapped in $\Re^3$, the appropriate choice is a discrete sets of harmonic weights, suggested by Dong [14] and is the one used in this paper (see Equation 2):

$$W_{ij} = \frac{1}{2}(\cot \alpha_{ij} + \cot \beta_{ij}). \tag{2}$$

Here α_{ij} and β_{ij} are the opposite angles to the edge (i, j).

Representing the function f, by the column vector of its values at all vertices $f = [f_1, f_2, \ldots, f_n]^T$, one can reformulate the Laplacian as a matrix $\Delta f = -Lf$ where the Laplacian matrix L each elements are defined by:

$$L_{ij} = \begin{cases} \sum_k W_{ik} & \text{if } i = j, \\ -W_{ij} & \text{if } (i, j) \text{ is an edge of } S, \\ 0 & \text{in other case.} \end{cases} \tag{3}$$

where k is the number of neighbors of the vertex i. The Eigenvalues $\lambda_1 = 0 \leq \lambda_2 \leq \ldots \leq \lambda_n$ of the matrix L forms the *spectrum* of mesh S. Besides describing the square of the frequency and the corresponding eigenvectors $e_1, e_2, \ldots, e_n$ of L, one can define piecewise linear functions over S using progressively higher frequencies [6].

3 Literature Review

A wide gamut of algorithms for surface reconstruction have been proposed in the literature in recent years [3] [7] [1].

Loop [5] generates B-spline surfaces on irregular meshes. These meshes do not require a known object topology, and therefore, they can be configured arbitrarily without carrying a sequence of the 3D coordinates of the points set. The advantage of this method is that it uses different spline types for the surface approximation. The algorithm was tested using synthetic data with low curvature.

Eck and Hoppe [11] present the first solution to the fitting problem of B-spline surfaces on arbitrary topology surfaces from disperse and unordered points. The method builds an initial parametrization, which in turn is re-parametrized to build a triangular base, which is then used to create a quadrilateral domain. In the quadrilateral domain, the B-spline patches adjust with a continuity degree of C^1. This method, although effective, is quite complex due to the quantity of steps and process required to build the net of B-spline patches on the adjustment surface.

Krishnamurthy and Levoy [15] presented a novel approach to adjust NURBS surface patches on cloud of points. The method consists of building a polygonal mesh on the points set first. Then on this mesh, a re-sampling is performed to generate a regular mesh, on which NURBS surfaces patches can be adjusted. The method has poor performance when dealing with complex surfaces. Other limitations are the impossibility to apply the method to surfaces having holes, and the underlying difficulty to keep continuity on the NURBS surface patches.

Park [8] proposed a two-phase algorithm. In the first phase, a grouping of the points is performed by means of the k-means algorithm to create a polyhedral mesh approximation of the points, which is later reduced to a triangular mesh, on which a quadrilateral mesh is built. In the second phase, the initial model is used to build a net of NURBS patches with continuity C^1. Park's proposal assumes that the cloud-of-points is closed in such a way that the NURBS patches network is fully connected. This implies that the proposed method is not applicable to open surfaces. The use of NURBS patches implies an additional process keeping continuity at the boundary, making the method computationally expensive even when the irregularity of the surface does not require it.

Boulanger *et al.* [12] describe linear approximation of continuous pieces by means of trimmed NURBS surfaces. This method generates triangular meshes which are adaptive to local surface curvature. First, the surface is approximated with hierarchical quadrilaterals without considering the jagged curves. Later, jagged curves are inserted and hierarchical quadrilaterals are triangulated. The result is a triangulation which satisfies a given tolerance. The insertion of jagged curves is improved by organizing the quadrilaterals' hierarchy into a *quad-tree* structure. The quality of triangles is also improved by means of a Delaunay triangulation. Although this method produces good results, it is restricted to surfaces which are continuous and it does not accurately model fine details, limiting its application for objects with an arbitrary topology.

Gregorski [4] proposes an algorithm which decomposes a given point-set into a data structure *strip tree*. The *strip tree* is used to adjust a set of minimal squares quadratic surfaces to the points cloud. An elevation to bi-cubic surfaces is performed on the quadratic surfaces, and they are merged to form a set of B-spline surfaces which approximates the given point-set. This proposal can not be applied to closed surfaces or surfaces which curve themselves. The proposal is highly complex because it has to perform a degree elevation and a union of patches on B-spline patches at the same time that a continuity degree C^1 is performed among adjacent patches.

Bertram [10] proposes a method to approximate in an adaptive way to disperse points by using triangular hierarchical B-splines. A non-uniform distribution of sampling on the surface is assumed, in such a way that zones with a high curvature present a denser sampling that zones with a low curvature. This proposal uses patches for data adjustment which add quality to the solution.

A different approach is presented by Yvart *et al.* [2], which uses triangular NURBS for dispersed points adjustment. Triangular NURBS do not require that the point-set has a rectangular topology, although it is more complex that NURBS. Similar to previous works, it requires intermediate steps where triangular meshes are reconstructed, re-parametrization processes are performed, and continuity patches G^1 are adjusted to obtain a surface model.

4 Approximation of Smooth Surfaces Using Morse Theory

The majority of the literature on re-meshing methods, focuses on the problem of producing well formed triangular meshes. However, the ability to produce quadrilateral meshes is of great importance as it is a key requirement to fit NURBS surface on a large 3–D mesh. Quadrilateral topology is the preferred primitives for modelling many objects and in many application domains. Many formulations of surface subdivision such as SPLINES and NURBS, require complex quadrilateral bases. Recently, methods to automatically

quadrilateralize complex triangulated mesh have been developed such as the one proposed by Dong *et al.* [14].

In this section, a method for the surface approximation by means of optimized NURBS patches from complex quadrilateral bases on triangulated surfaces of arbitrary topology is proposed. This process of quadrilateralization produces regions composed exclusively of smooth quadrilaterals. To decompose the triangulated surface into quadrilateral patches, Morse theory and spectral mesh analysis are used. The quadrilateral border joining the critical points are regularized by computing geodesic curves between each corner and then B-splines approximate those geodesics. Following the geodesic curves approximation a NURBS surface is then fitted by changing the NURBS's weight to represent the data inside the quadrilateral region. Such NURBS surfaces fitting is non-linear and an evolutionary strategy optimization method is used to minimize the distance between the surface and the points inside the quadrilateral region. The optimization also takes into account the smooth joint at the boundary to guarantee C^1 continuity.

4.1 Quadrilateralization of Triangular Mesh

One of the first step of our algorithm consist of converting a triangular representation into a network of quadrilateral that is a complete description of the object's geometry. This is necessary as the representation by means of NURBS patches requires building a regular base on which the NURBS surfaces sits. Because of the complex and diverse forms of free-formed objects, obtaining a quadrilateral description of the whole surface is not a trivial task.

4.1.1 Localizing Critical Points

Initially, the quadrilateral's vertices are obtained as a critical point-set of a Morse function. Morse's discrete theory guarantees that, without caring about topological complexity of the surface represented by triangular mesh, a complete quadrilateral description is obtained. That is to say, it is possible to completely divide objects' surfaces by means of rectangles. In this procedure, an equation system for the Laplacian matrix is solved by calculating a set of eigen-values and eigen-vectors for each matrix (Equation 3) [16].

A Morse-Smale complex is obtained from the connection of a critical point-set which belongs to a field of the Laplacian matrix. The definition of a field of the matrix is obtained by selecting the set of vectors associated to a solution value of the equation. As Morse function represents a function in the mesh, each eigen-value describes the frequency square of each function. Thus, selecting each eigen-value directly indicates the quantity of critical points which the function has. For higher frequency values, a higher number of critical points will be obtained. This permits representing each object with a variable number of surface patches. The eigen value computations assigns function values to every vertex of the mesh, which permits determining whether a vertex of the mesh is at critical points of the Morse function. In addition, according to a value set obtained as the neighborhood of the first ring of every vertex, it is possible to classify the critical points as maximum, minimum or "saddle points." Identification and classification of every critical point permits building the Morse-Smale complex.

4.1.2 Critical Points Interconnection

Once critical points are obtained and classified, then they should be connected to form the quadrilateral base of the mesh. The connection of critical points is started by selecting a "saddle point" and by building two inclined ascending lines and two declined descending lines. Inclined lines are formed as a vertex set ending at a maximum critical point. In addition, a descending line is formed by a vertex path which ends at a minimum critical point. One can then join two paths if both are ascending or descending.

After calculating every paths, the triangulation of K surface is divided into quadrilateral regions which forms Morse-Smale complex cells [16]. Specifically, every quadrilateral of a triangle falls into a "saddle point" without ever crossing a path. The complete procedure is described in Algorithm 1.

Algorithm 1: Bulding method of MS cells.

```
Critical points interconnection();
begin
    Let T={F,E,V} M triangulation;
    Initialize Morse-Smale complex, M=0;
    Initialize the set of cells and paths, P=C=0;
    S=SaddlePointFinding(T);
    S=MultipleSaddlePointsDivission(T);
    SortByInclination(S);
    for every s ∈ S in ascending order do
        CalculeteAscedingPath(P);
    end
    while exists intact f ∈ F do
        GrowingRegion(f, p0, p1, p2, p3);
        CreateMorseCells(C, p0, p1, p2, p3);
    end
    M = MorseCellsConnection(C);
end
```

4.2 Regularization of the Quadrilateral Border Curves

Because the surface needs to be fitted using NURBS patches, it is necessary to regularize the quadrilateral curves obtained from the mesh. The curves are regularized and fitted by b-splines using the following Algorithm 2.

Algorithm 2: Quadrilateral mesh regularization method..

```
Regularization();
begin
    1. Quadrilateral selection;
    2. Selection of a border of the selected quadrilateral and its opposite;
    3. Regularization using B-splines with lambda density;
    4. Regularized points match by means of geodetics FMM;
        4.1 Smoothing of geodetic with B-splines;
    5. Points generating for every B-spline line with lambda density;
end
```

One of the quadrilateral border is selected from the mesh, and later a border is selected from each quadrilateral border and its opposite. The initially selected border is random. The opposite order is searched as one which does not contain the vertices of the first one. If the first selected border has vertices A and B, it is required that the opposite border does not contain vertices A and B, but the remaining, B and C.

Later, B-splines are fitted on selected borders with a λ density, to guarantee the same points for both borders are chosen, regardless of the distance between them. In general, a B-spline does not interpolate every control point; therefore, they approximate curves which permit a local manipulation of the curve, and they require fewer calculations for coefficient determination.

Having these points at selected borders, it is required to match them. This is done with FMM (*Fast Marching Method*). This algorithm is used to define a distance function from an origin point to the remainder or surface with a computational complexity of $O(n \times \log n)$. This method integrates a differential equation to obtain the geodetic shortest path by traversing the triangle vertices.

At the end of the regularization process, B-splines are fitted on geodetic curves and density λ points are generated at every curve which unite the border points of quadrilateral borders, to finally obtain the grid which is used to fit the NURBS surface.

4.3 Fitting of Optimized NURBS Patches Using an Evolutionary Algorithm

This section presents a method based on an evolutionary strategy (ES), to determine the weights of control points of a NURBS surface, without modifying the location of sampled points of the original surface. The main goal is to reduce the error between the NURBS surfaces and the data points inside the quadrilateral regions. In addition, the algorithm make sure that the C^1 continuity condition is preserved for all optimized NURBS patches. The proposed algorithm is described in Algorithm 3.

Algorithm 3: Optimization and continuity method of NURBS patches method.

```
Adjustment by optimized NURBS patches();
begin
    1. Optimization of the NURBS patches;
        1.1. Multiple ES usage with deterministic replacement by inclusion;
        1.2. Application of ES to control weights of NURBS;
    2. Union of NURBS patches with continuity C^1;
        2.1. Check continuity between axis;
        2.2. Check continuity at vertices;
end
```

4.3.1 Optimization of NURBS Parameters

A NURBS surface is completely determined by its control points $\mathbf{P}_{i,j}$. The main difficulty in fitting NURBS surface locally is in finding an adequate parametrization for the NURBS and the ability to automatically choose the number of control points and their positions. The NURBS's weight function $w_{i,j}$ determine the local influence degree of a point in surface topology. Generally, weights of control points for a NURBS surface are assigned in an homogeneous way and are set equal to 1, reducing NURBS to simple B-spline surface. The determination of NURBS control points and weights for arbitrarily curved surfaces adjustment is a complex non-linear problem.

The optimization process is formally described as follows: Let $P = \{p_1, p_2, \ldots, p_n\}$ be a set of 3-D points sampled from a real object, which has rectangular topology, and $S = \{s_1, s_2, \ldots, s_m\}$ be a NURBS surface that approximates P, our problem consist of minimizing the approximation error given by 4

$$E(S) = d_{P,S} < \delta \tag{4}$$

where $d_{P,S}$ is the total distance between P and the NURBS approximation surface S. The parameter δ is a given user error tolerance. It is attempted obtain the configuration of S so that 4 is true.

Since the influence of the NURBS surface control points is only local, the sampled points P will be divided in clusters where will carry on a local optimization process, which reduces the computational cost of the proposed method.

The optimization process starts with a clustering of the set of points P such clustering will be achieved by a SOM. The objective of the SOM is to find homogeneous regions where run the optimization process without distort the local shape of the surface. The points of P will be presented to the SOM as the training patterns. It is hoped at last of the training the SOM have found the homogeneous regions where run the optimization process.

Once clustered P an evolutionary strategy $(\mu + \lambda) - ES$ will optimize the local fitting of the NURBS in each cluster. The evolutionary strategy configuration is as follow:

Individuals: the individuals of the strategy are conformed by the weights of the cluster points and the mutation steps σ, like shows Figure 1.

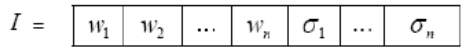


Figure 1: Individual of the strategy

where w_i are the control point weights and σ_i are the mutation step sizes.

Mutation operator: uncorrelated mutation with n mutation step sizes σ's is applied to the individuals.

Recombination operator: the recombination operator is different for object variable w_i than parameters σ_i. A global intermediary recombination is applied to object variables, according to 5, whereas a local intermediary recombination is applied to mutation step sizes σ_i, according to 6.

$$b_i' = \frac{1}{\rho}\sum_{k=1}^{\rho} b_{k,i} \tag{5}$$

$$b_i' = \mu_i b_{k_1,i} + (1-\mu_i) b_{k_2,i} \tag{6}$$

where i is the allele of the individual, b_i is the value of the allele, ρ is the size of the recombination pool and μ is a random number uniformly distributed in $[0,1]$.

Selection operator: the best individuals according to the aptitude function given in 4. In order to perform a fast compute of the distance between the points P and the NURBS surface S, the points of S are store in a kd-tree structure, so that the searching process for finding the nearest points between P and S is $\log(n)$ order. The Algorithm 4 summarizes the optimization process.

Algorithm 4: Perform a clustering of P by using SOM.

```
begin
    for each cluster do
        Set individual size = cluster size;
        Set population size = μ;
        Initialize randomly the population;
        Evaluate the population in the aptitude function 4;
        while the stop criterion δ do not reached do
            for i = 1 to λ · 0.9 do
                Ind_i = mut(Population_rand(1,μ));
            end
            for i = 1 to λ · 0.1 do
                Ind_i = rec(Population_rand(1,μ));
            end
            Population = select from(μ + λ);
        end
    end
end
```

4.3.2 NURBS Patches Continuity

Continuity in regular cases (4 patches joined at one of the vertex) is a solved problem [11]. However, in neighborhoods where the neighbors' number is different from 4 ($v \geqslant 3 \rightarrow v \neq 4$), continuity must be adjusted to guarantee a soft transition of the implicit surface function between patches of the partition. In this paper, C^1 continuity between NURBS patches is guaranteed, using Peters continuity model [9] which guarantees continuity of normals between bi-cubical spline functions. Peters proposes a regular and general model of bi-cubical NURBS functions with regular nodes vectors and the same number of control points at both of the parametric directions. In our algorithm, Peter's model was adapted by choosing generalizing

NURBS functions, with the same control points number at both of the parametric directions, bi-cubic basis functions and regular expansions in their node vectors.

Continuity Along the Quadrilateral Boundaries: To guarantee C^1 continuity between the boundaries of neighboring patches, extreme control points which affect the continuity between patches must be found. Due to data ordering within the proposed parametrization schema, two adjacent patches will have the same number of control points at the common axis, regardless of their disposition. To adjust continuity between axes, control points are calculated on the analyzed boundary, to make it co-lineal with neighboring control points on adjacent patches.

Equation 7 illustrates the new position for a control point at given P_{eje} axis, where P_A^{vec} is the neighbor point to P_{eje} at patch B. The new control point P_{eje} is the medium point between the two adjacent control points P_A^{vec} y P_B^{vec} which guarantees that control points on the axis and their adjacent neighbors at each patch is co-linear.

$$P_{eje} = \frac{P_A^{vec} + P_B^{vec}}{2} \tag{7}$$

Continuity at Quadrilateral Vertices: Continuity at vertices of quadrilateral regions is guaranteed by making sure that every adjacent control points at each vertices is co-planar.

Under the continuity criteria proposed by Peters, continuity at quadrilateral vertices are generalized, that is to say, the adjustment process is the same regardless of the number of patches which can be found at a given vertex. We have $\pi^T P = 0$, where π is a given plane and P is a point on the plane. If the system of equations is over-determined with more than four points, the equation which best adjusts a given point-set can be found.

Equation 8 represent the over-determined system where $P = [P_1, P_2, \ldots, P_n]^T$ with $n \geq 4$ are control points at the vertices. The equation is solved using Singular Values Decomposition *SVD*, with the last column of matrix P the equation of the plane which is adjusted to point-set P in the quadratic mean square error sense. [13].

$$\begin{bmatrix} P_x^1 & P_y^1 & P_z^1 & 1 \\ P_x^2 & P_y^2 & P_z^2 & 1 \\ \ldots & \ldots & \ldots & 1 \\ P_x^n & P_y^n & P_z^n & 1 \end{bmatrix} \begin{bmatrix} \pi_x \\ \pi_y \\ \pi_z \\ \pi_z \end{bmatrix} = 0 \tag{8}$$

Continuity is adjusted by projecting control points P onto the plane given by Equation 8:

$$\pi = n_1(x - x_o) + n_2(y - y_o) + n_3(z - z_o) \tag{9}$$

where $N = [n_1, n_2, n_3]$ is the plane's normal and $P_0 = [x_0, y_0, z_0]$ is a point on the plane. The projection $P_I = [x_I, y_I, z_I]$ of a point $P = [P_x, P_y, P_z]$ on the plane is given by:

$$x_I = P_x + n_1 t_I \quad y_I = P_y + n_2 t_I \quad z_I = P_z + n_3 t_I \tag{10}$$

where t is the parametric value of the straight line which passes through point P in the direction of the plane's normal **N**.

Using Equation 10, it is possible to project control points on the given plane, which guarantee the continuity of normals at vertices of the quadrilateral partition, ensuring that every adjacent control points are co-planar.

5 Experimental Results

Tests were performed using a 3.0 GHz dual Opteron processor computer, with 1.0 GB RAM, running Microsoft Windows XP operating system. The methods were implemented using C++ and MATLAB. The data used were obtained with Kreon range scanners, available at the Advanced Man-Machine Laboratory – Department of Computing Science, University of Alberta, Canada.

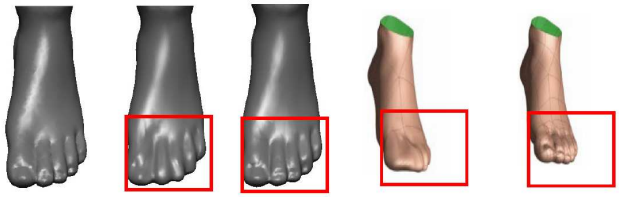


Figure 2: Comparison Between Branch's Method and Eck and Hoppe's Method. a) Triangulated model, b) 27 patches model (Branch's method without optimize), c) 27 patches model (Branch's method optimized), d) 29 patches model (Eck and Hoppe's method without optimize), e) 156 patches model (Eck and Hoppe's method optimized)

5.1 Comparison Between Branch's Method and Eck and Hoppe's Method

The work by Eck and Hoppe [11] performs the same adjustment by means of a network of B-spline surface patches adaptatively refined until they obtain a given error tolerance. The process of optimization performed by Eck and Hoppe reduces the error by generating new patches, which considerably augments the number of patches which represent the surface. The increment of the number of patches reduces the error because the regions to be adjusted are smaller and more geometrically homogeneous. In the method proposed in this paper, the optimization process is focused on improving the adjustment for every patch by modifying only its parameterization (control points weight). Because of that, the number of patches does not augment after optimization process. The final number of patches which represent every object is determined by the number of critical points obtained in an eigenvector associated with the eigenvalue (λ) selected from the solution system of the Laplacian matrix, and it does not change at any stage of the process.

Figure 2 contains a couple of objects (foot and skidoo) reported by Eck and Hoppe. Every object is shown triangulated starting with the points cloud. The triangulation is then adjusted with a patch cloud without optimizing and the result obtained after optimization. The adjustment with the method proposed in this paper, represents each object, with 27 and 25 patches, while Eck and Hoppe use 156 and 94 patches. This represents a reduction of 82% and 73% fewer patches respectively, in our work.

With respect to the reduction of the obtained error in the optimization process in each case, with the proposed method in this paper, the error reduces an average of 77%, a value obtained in an experimental test with 30 range images. Among these appear the images included in Figure 2. The error reported in Eck and Hoppe for the same images of Figure 2 allow a error reduction of 70%. In spite of this difference which is given between our method with respect to Eck and Hoppe's method, we should emphasize that error metrics are not the same, Eck and Hoppe's method is a measurement of RMS, ours method corresponds to an average of distances of projections of points on the surface.

Another aspect to be considered in the method comparison is the number of patches required to represent the object's surfaces. In Eck's work, the number of patches used to represent the object's increase is an average of 485% in relation to the initial quadrilaterization, while in the method proposed in this paper, the number of patches to represent the surface without optimization, and the optimized one, is constant.

6 Conclusion

The methodology proposed in this paper for the automation of reverse engineering of free-form three-dimensional objects has a wide application domain, allowing adjustment of surfaces regardless of topological complexity of the original objects.

A novel method for fitting triangular mesh using optimized NURBS patches has been proposed. This method is topologically robust and guarantees that the complex base be always quadrilateral creating a network of surfaces which is compatible with most commercial CAD systems.

In the proposed algorithm, the NURBS patches are optimized using multiple evolutionary strategies to estimate the optimal NURBS parameters. The resulting NURBS are then joined, guaranteing C^1 continuity. The formulation of C^1 continuity presented in this paper can be generalized, because it can be used to approximate regular and irregular neighborhoods which present model processes regardless of partitioning and parametrization.

References

[1] Arge A. Approximation of scattered data using smooth grid functions. *Comp. App.l. Math.*, 59:191–205, 1995.

[2] Yvart A., Hahmann S., and Bonneau G. Smooth adaptive fitting of 3-d models using hierarchical triangular splines. pages 13–22, Boston, USA, 2005.

[3] Baxter B. *The Interpolation Theory of Radial Basis Functions*. PhD thesis, Trinity College, University of Cambridge, UK, 1992.

[4] Gregorski B., Hamann B., and Joy D. Reconstruction of b-spline surfaces from scattered data points. pages 163–170, Geneva, Switzerland, June 2000.

[5] Loop C. Smooth spline surfaces over irregular meshes. pages 303–310, Orlando, USA, July 1994.

[6] Weber G., Scheuermann G., Hagen H., and Hamann B. Exploring scalar fields using critical isovalues. pages 171–178, Boston, USA, October 2002.

[7] Hoppe H. *Surface Reconstruction From Unorganized Points*. PhD thesis, Washington University, USA, 1994.

[8] Park I., Lee S., and Yun I. Constructing nurbs surface model from scattered and unorganized range data. pages 312–320, Ottawa, Canada, October 1999.

[9] Peters J. Constructing c^1 surfaces of arbitrary topology using bicuadric and bicubic splines. *Designing Fair Curves and Surfaces*, 277-293, 1994.

[10] Bertram M., Tricoche X., and Hagen H. Adaptive smooth scattered-data approximation for large-scale terrain visualization. *Proc. Symp. on Data Visualisation*, pages 177–184, May 2003.

[11] Eck M. and Hoppe H. Automatic reconstruction of b-spline surface of arbitrary topological type. *Proc. 23rd International Conference on Computer Graphics and Interactive Techniques*, pages 325–334, August 1996.

[12] Boulanger P. Triangulating nurbs surfaces, curve and surface design. Technical report, Vanderbilt University Press, Nashville, Tennesee, USA, 2000.

[13] Hartley R. and Zisserman A. *Multiple view geometry in computer vision*. Cambridge University Press, second edition edition, 2003.

[14] Dong S., Bremer P., Garland M., Pascucci V., and Hart J. Quadrangulating a mesh using laplacian eigenvectors. Technical report, University of Illinois, USA, 2005.

[15] Krishnamurthy V. and Levoy M. Fitting smooth surfaces to dense polygon meshes. *Proc. 23rd International Conference on Computer Graphics and Interactive Techniques*, pages 313–324, August 1996.

[16] Ni X., Garland M., and Hart J. Fair morse functions for extracting the topological structure of a surface mesh. *ACM Trans. Graph.*, 23(3):613–622, 2004.

[17] Ni X., Garland M., and Hart J. Simplification and repair of polygonal models using volumetric techniques. *Proc. SIGGRAPH*, TOG 23,3,613-622, 2004.

Fully Automated Laser Range Calibration

Matthew Antone and Yuli Friedman
Computer Vision Group
BAE Systems Advanced Information Technologies
Burlington, MA, USA
{matthew.antone,yuli.friedman}@baesystems.com

Abstract

We present a novel method for fully automated exterior calibration of a 2D scanning laser range sensor that attains accurate pose with respect to a fixed 3D reference frame. This task is crucial for applications that attempt to recover self-consistent 3D environment maps and produce accurately registered or fused sensor data.

A key contribution of our approach lies in the design of a class of calibration target objects whose pose can be reliably recognized from a single observation (i.e. from one 2D range data stripe). Unlike other techniques, we do not require simultaneous camera views or motion of the sensor, making our approach simple, flexible and environment-independent.

In this paper we illustrate the target geometry and derive the relationship between a single 2D range scan and the 3D sensor pose. We describe an algorithm for closed-form solution of the 6 DOF pose that minimizes an algebraic error metric, and an iterative refinement scheme that subsequently minimizes geometric error. Finally, we report performance and stability of our technique on synthetic and real data sets, and demonstrate accuracy within 1 degree of orientation and 3 cm of position in a realistic configuration.

1 Introduction

In recent years, the ubiquity of laser range sensors (*lidars*) has increased, and their application to many domains – including vision and robotics – has grown rapidly. Indeed, heterogeneous sensor suites consisting of multiple lidars, cameras, and other devices have become quite common for such applications as object recognition, tracking, navigation, and environment reconstruction.

For multi-sensor systems to be useful, they must produce measurements in a common coordinate system so that observations from two or more sensors may be related to one another in a meaningful way within a consistent *relative* reference frame. Further, for higher-level geometric reasoning it is often essential for sensors to be situated with respect to a known *absolute* reference frame. In simultaneous localization and mapping, for instance, sensors must be carefully calibrated with respect to the robot's body frame to enable the robot to accurately localize obstacles and landmarks (Figure 1).

Several different types of lidar devices currently exist, with varying scanning mechanisms, number of lasers, and geometric configurations. In this work we focus on the most

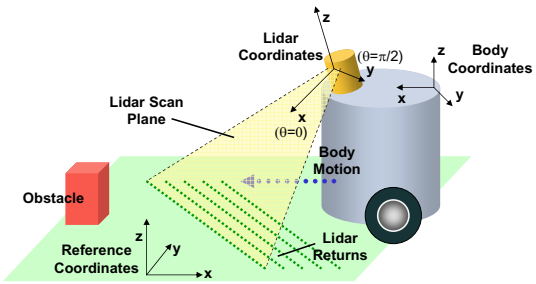


Figure 1: A typical application in which a lidar is rigidly attached to a mobile robot that produces pushbroom-like scans of its environment. We wish to determine the transformation between the lidar's coordinate system and that of a known reference such as the body frame or the environment.

common, the planar or line-scanning lidar. The calibration of absolute orientation is especially important and challenging for these sensors, as they produce only 2D slices of their 3D environment. As the lidar plane moves (e.g. spins at a fixed location to produce full spherical scans, or translates on a mobile robot to sweep out the environment ahead), systems require accurate hand-eye calibration to assemble the individual slices into a metric – and thus physically meaningful – 3D point cloud.

In this paper we present a novel and fully-automated procedure for calibrating the exterior 6-degree-of-freedom pose, consisting of 3D location and orientation, for a planar lidar sensor with respect to a fixed reference frame. We make no assumptions about the surrounding environment and do not require the sensor to physically move. Our algorithm merely requires a single range-only scan of a carefully-designed calibration target, making the technique flexible and independent of errors in other sensor measurements such as odometers or servos.

1.1 Prior Work

There are many strategies for calibrating the pose of a lidar sensor. In the literature, these tend to fall into two main categories: those that require the use of additional sensors such as cameras, and those that require known motion of the lidar itself. Multi-sensor calibration typically determines the relative Euclidean transformation between the lidar and a rigidly-attached camera. Zhang and Pless present a technique in which both sensors image a planar checkerboard target at different (unknown) orientations; the camera's pose with respect to the target is determined using a standard extrinsic calibration method, and combined with straight-line profiles extracted from the lidar to form constraints on the lidar's pose [10]. Mei and Rives developed a more general theory of lidar registration to catadioptric cameras, considering additional cases in which the laser returns are visible to the camera, thus forming explicit correspondences between image pixels and range samples [6]. Several methods have also been proposed that utilize structured light, or visible laser profiles in the images [4, 7].

Motion-based or *active* calibration involves imaging objects from multiple locations and orientations. Several methods move a robotic arm, to which the laser is attached, and

capture range data of a fixed planar surface at different orientations; knowledge of precise relative arm pose is assumed [8, 2]. McIvor places a cubic calibration target on a motion table, and "scans" the object with the lidar to obtain both range and intensity data; the laser pose that best rectifies the data to the known target geometry is then determined [5]. Finally, Zhang and Pless estimate egomotion of a moving robot via robust matching of consecutive lidar scans over time, using the structure of the surrounding environment to constrain the hand-eye lidar pose [9].

1.2 Contributions

In this work, we develop a flexible lidar calibration technique based on a novel target object design. Our technique has several unique characteristics that provide advantages over prior methods: it does not require additional sensors such as cameras; it does not require motion of the sensor or the target; and it requires only range data, not intensity data. These properties offer applicability to a wide variety of different 2D laser scanners and imaging geometries. Further, because our target's surface can be painted with visible patterns, our method also enables precise lidar-camera registration. To our knowledge, this is the first such published technique.

2 Fundamentals

We seek the rigid-body Euclidean transformation (R,T) that aligns a particular sensor's local coordinate system with some fixed reference frame. Here, T is the sensor's 3D location in the reference frame, and the columns of the orthonormal rotation matrix R represent the axes of the sensor expressed in reference coordinates. Thus, the transformation between a reference point M and a sensor point Q is given by

$$Q = R^T(M-T), \tag{1}$$

and the inverse transformation is given by

$$M = RQ+T. \tag{2}$$

The laser produces 2D range scans consisting of discrete angular samples in a plane. For each ray sample along the direction denoted by angle θ_k, the sensor measures a corresponding distance r_k along that ray. We place the origin of the sensor coordinate system at the origin of all laser rays; the ray at $\theta = 0$ is coincident with the sensor's x axis, and the ray at $\theta = \pi/2$ is coincident with the y axis. Thus, the scan produces measurements of the form

$$Q_k^T = \begin{pmatrix} u_k & v_k & 0 \end{pmatrix} = r_k \begin{pmatrix} \cos\theta_k & \sin\theta_k & 0 \end{pmatrix}. \tag{3}$$

Since the range measurements all lie in a plane, the transformation between lidar coordinates and reference coordinates given in (2) can be reduced to a homography [3] as

$$M = Hq = RKq = R\begin{pmatrix} e_x & e_y & R^T T \end{pmatrix} q, \tag{4}$$

where e_x and e_y represent unit vectors in the x and y direction, respectively, and $q^T = \begin{pmatrix} u & v & 1 \end{pmatrix}$.

Internal mis-calibration and servo scan inconsistency may cause error in θ, which is generally negligible. Range measurements r also exhibit uncertainty induced by quantization (i.e. fixed bit allocation in analog-to-digital conversion of the range return), typically on the order of 1cm, and by measurement noise from material reflectivity and environmental effects, typically also on the order of 1cm [1]. Finally, nonlinear range errors can arise from depth discontinuities in imaged surfaces due to non-zero laser spot size.

3 Calibration Target Design

The most important factor in designing a target object suitable for single-scan range-based calibration is the 3D pose recovery itself. While many simple objects allow determination of certain straightforward extrinsic parameters, the design problem becomes more challenging as the number of required DOFs increases. A second factor is reliable and stable detection of the target object, which must account for lidar measurement errors and discrete rather than continuous angular sampling. Finally, there are practical matters to consider such as the ease, repeatability, and accuracy of physical target construction.

The lidar produces as measurements a series of ranges at specified angles that constitute the (sampled) intersection of a virtual plane with the visible 3D surfaces in the scene. We next consider two classes of simple 3D objects that produce unique 2D cross sections determined entirely by the orientation and position of the slicing plane.

3.1 Conic Sections

It is well known that conic sections are produced by intersections of a plane with a double cone, with the particular type and shape of the curve defined entirely by the position and orientation of the slicing plane. For simplicity, and without loss of generality we assume a cone with apex at the origin and whose axis of symmetry is the z axis in reference coordinates, defined by the implicit equation

$$X^2 + Y^2 - Z^2 = 0 = M^T S M, \tag{5}$$

where the reflection matrix $S = diag(1,1,-1)$. Intuitively, the size of a cross section is related to the translation of the sensor slice plane along the symmetry axis, and the shape of the cross section is related to the slice plane's orientation (Figure 2).

Substitution of (4) into (5) reveals

$$(Hq)^T S(Hq) = 0 = q^T K^T R^T S R K q. \tag{6}$$

Thus, each observed lidar sample generates a single quadratic constraint equation in the 6 unknown parameters encoded by R and T, and it would seem that six such points in non-degenerate configuration uniquely determine these parameters. However, due to inherent geometric symmetries, not all DOFs may in fact be recovered.

A geometric illustration of this ambiguity is shown in Figure 2. Algebraically, we can see that applying a rotation about z to reference point M does not affect (6) since the reflection and rotation axes are identical; we can nonetheless obtain an accurate estimate for R and T in closed form, modulo this rotation about z (see Appendix A). Unfortunately, this inherent ambiguity precludes use of the cone for full 6 DOF calibration. Further, a conic target is difficult to construct with sufficient size and precision for pose recovery.

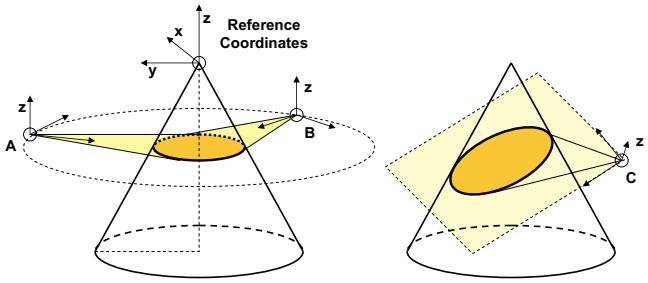


Figure 2: A conic section orthogonal to the symmetry axis forms a circle whose size varies with z. Lidars with poses A and B would produce the same circular scan (left). An off-axis slice plane produces elliptical or hyperbolic cross sections determined by the plane's orientation (right).

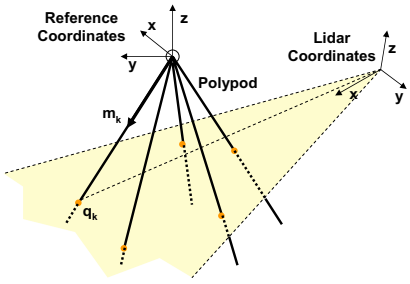


Figure 3: Polypod legs with direction m_k intersect the lidar slice plane to form points q_k in sensor coordinates. The relationship between m_k and q_k is a plane projective homography.

3.2 Polypods

We define a *polypod* as a multi-legged structure consisting of a set of rays that emanate from a single point in known directions. Without loss of generality we assume that the ray origin is coincident with the origin of the reference coordinate system. As with the cone, moving the slice plane along the z axis changes the size of the cross section, and tilting the plane changes the cross section's shape (Figure 3). However, unlike the cone, a polypod can be designed so that its cross section uniquely determines the sensor's full 6 DOF pose, up to a duality of solutions.

We now derive a geometric relationship between the polypod legs and the lidar measurements given a particular orientation R and position T. Let $m_k^T = \begin{pmatrix} x_k & y_k & 1 \end{pmatrix}$ represent the direction of the kth ray, or polypod leg; reference points M on the 3D ray may thus be parameterized as $M = m_k s$. To determine the lidar points Q, we then transform the ray to lidar coordinates and find its intersection with the lidar slice plane $z = 0$. The new ray direction is given by $R^T m_k$, and the new ray origin is $-R^T T$, according to the transformation (1). Thus, the ray parameterization becomes

$$Q = -R^T T + R^T m_k s. \tag{7}$$

To find the intersection of the ray and the plane $z = 0$, we determine the parameter s such that the z component of Q is zero. Solving (7) for $Q^T e_z = 0$ gives $s = R_z^T T / R_z^T m_k$, where R_z is the third column of the rotation matrix R. Substituting back into (7) gives

$$Q = R^T (m_k \frac{R_z^T T}{R_z^T m_k} - T), \tag{8}$$

which we can re-write component by component as

$$\begin{pmatrix} u_k \\ v_k \\ 1 \end{pmatrix} R_z^T m_k = q_k R_z^T m_k = \begin{pmatrix} R_x^T m_k R_z^T T - R_x^T T R_z^T m_k \\ R_y^T m_k R_z^T T - R_y^T T R_z^T m_k \\ R_z^T m_k \end{pmatrix}. \tag{9}$$

After manipulation of (9), we find that

$$q_k \sim \begin{pmatrix} t_z & 0 & -t_x \\ 0 & t_z & -t_y \\ 0 & 0 & 1 \end{pmatrix} R^T m_k = K^{-1} R^T m_k \tag{10}$$

where $t = R^T T$. The 3×3 matrix $K^{-1} R^T$ defines a plane projective homography between measured points q_k on the lidar slice plane and the corresponding polypod rays m_k.

Note that (10) is the inverse of (4), the basic relationship between lidar points and reference points; here, polypod legs equate to projective rays. Further, we observe that the ambiguity from symmetry in the conic case is completely resolved, because cross sections are not rotationally symmetric and because there is a unique homography that relates points to points, rather than points to surfaces, provided that at least four non-degenerate correspondences exist [3].

3.3 Pyramid Target

Having derived the abstract geometric relationship between polypod configuration and lidar cross section, we now define a more practical and concrete calibration target design. First, for precise and simple construction, and because a homography is uniquely defined by four correspondences, we use the minimum number of four legs. We also avoid construction of a literal polypod; noisy measurements of its thin legs would lead to significant errors in recovered pose, and in fact the legs might be missed by the laser entirely.

Rather than detecting the polypod legs directly as single-point measurements, then, we design our target so that we can indirectly, but more reliably, infer their locations in the scan. To achieve this, we construct a pyramid-shaped target whose four planar faces intersect to form "virtual" polypod legs (Figure 4). The cross section is a quadrilateral whose edges may be estimated directly from sets of multiple scan points, and whose vertices q_k may then be recovered more precisely as the intersections of these edges.

4 Pose Recovery

To reduce the effects of quantization error and stochastic noise in range and angle measurements, we keep the target stationary in the lidar's field of view for a few seconds and

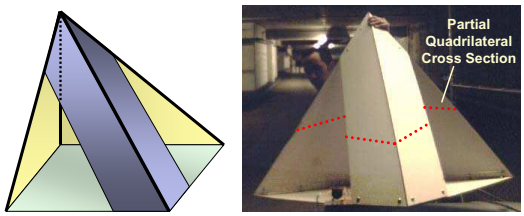


Figure 4: Our target design consists of a pyramid (left) with holes cut into the front faces so that the back faces are visible. Bold lines indicate "virtual" polypod legs formed by intersection of the pyramid faces. A lidar scan of this object (right) forms a quadrilateral cross section that uniquely determines the sensor pose.

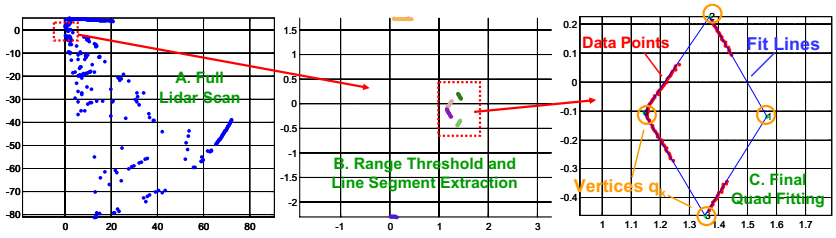


Figure 5: Target segmentation involves several steps that result in rejection of clutter points and a robustly estimated quadrilateral cross section of the target object.

obtain a series of several hundred scans, which are then averaged to form a single, noise-suppressed scan. Our algorithm processes this data to automatically segment the target points from the background, find the vertices of the quadrilateral cross section, solve for the approximate pose in closed form, and iteratively refine the pose for higher accuracy.

4.1 Segmentation

A particular lidar scan contains both the desired cross section of the calibration target object and clutter from the environment. Our first task is therefore to segment the points of interest from the clutter (Figure 5). We first remove all points beyond a threshold distance from the sensor. We next search the scan for contiguous straight line segments using a successively applied RANSAC algorithm [3]. Each RANSAC iteration selects two points at random from the scan, fits a line to those points, and evaluates the remaining points against the line. The line associated with the most inliers is kept, the inliers are removed from the scan points, and the procedure repeats until no additional lines are found.

We next subdivide each line into contiguous segments, removing segments shorter than a threshold length, where the threshold is related to the target's expected cross sectional dimensions. The final step is to find the "correct" four line segments corresponding to the true target cross section. We first search for candidate segment pairs that might form the front two faces; we then search the remaining segment pairs for closed quadrilaterals.

4.2 Position and Orientation Estimation

We form the four vertices q_k via intersection of the extracted line segments, then apply an ordering constraint to assign the q_k to the correct (known) polypod leg directions m_k. Having established four correspondences, we solve for the homography $H = RK$ using a standard technique [3].

We next factor H using QR decomposition, which results in an orthonormal matrix R and an upper-triangular $\tilde{K}$. Because of the unknown scale on H, the matrix $\tilde{K}$ will not be of the proper form of (4). In the absence of noise, it will differ by a constant scale factor, so we divide $\tilde{K}$ by either of its first two diagonal entries (which are equal) and determine T by left-multiplying the third column of the result by $-R$. In general, however, the q_k will be noisy, and $\tilde{K}$ will not simply differ from K by a constant scale factor. We therefore approximate the true K by dividing $\tilde{K}$ by the average of its first two diagonal entries (which are no longer equal), and determining T as before. Note that this differs from the factorization of Zhang [11], which solves for K exactly and then approximates R by finding the "closest" rotation in Frobenius norm to the estimated $\tilde{R}$.

In general, when noise is present in the data, the above algebraic approximation leads to a pose solution that is reasonably accurate, but not optimal. This solution can, however, be used to initialize a direct optimization method on R and T, such as Levenberg-Marquardt, that seeks to minimize the sum of Euclidean distances between the transformed points Hq_k and their counterparts m_k.

5 Experiments

To demonstrate the efficacy of our target object design and calibration method, we conducted several different experiments in simulation (to systematically control noise sources and other DOFs) and on real data (to show performance with physical sensor, target, and measurements). Results of these experiments are presented below.

5.1 Synthetic Data

We generated simulated data for experiments as follows. First, we constructed a virtual 3D model of the target pyramid with realistic dimensions (approximately one meter on a side). Next, we chose a set of lidar poses with translations varying between 2 and 30 meters and rotations varying between 0 and 45 degrees about each axis. For each pose we generated virtual lidar scans by intersecting scan rays with the target surface at 0.25 degree spacing. Ray angles and intersection ranges were perturbed by additive Gaussian noise with specified variance; ranges were also quantized to 1cm levels, and range discontinuities were averaged to simulate real sensor phenomena mentioned in Section 2. Results are shown in Figure 6; we report the mean position and orientation errors over 500 trials for each variant.

5.2 Real Data

We constructed a physical calibration target (Figure 4) and configured six different sensor poses with the target sitting on the ground. We measured the lidar's position and orientation by hand, so the "ground truth" pose was known only approximately. The sensor was

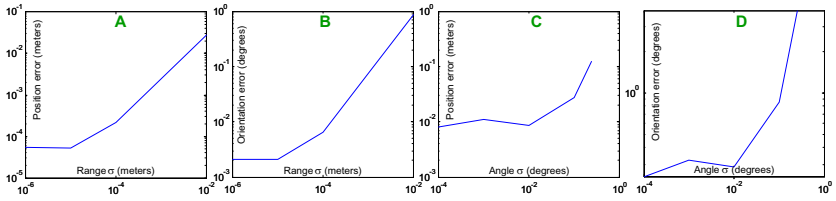


Figure 6: Performance of our algorithm on simulated data, plotting position and orientation error with varying range uncertainty (A and B) and angular uncertainty (C and D).

Table 1: Position and Rotation Error using Real Data

Dist (cm)	Roll (°)	Pitch (°)	Yaw (°)	Pos Err (cm)	Rot Err (°)
356 ± 5	0 ± 2	0 ± 2	30 ± 2	3.1	1.1
311 ± 5	0 ± 2	30 ± 2	15 ± 2	5.4	1.8
347 ± 5	10 ± 2	0 ± 2	45 ± 2	4.9	0.9
312 ± 5	10 ± 2	30 ± 2	45 ± 2	6.1	1.3
623 ± 5	10 ± 2	30 ± 2	45 ± 2	2.2	0.5
988 ± 5	10 ± 2	30 ± 2	45 ± 2	3.2	0.4

a SICK LMS291 set for 0.25° angular and 1cm range resolution. At each pose, we collected and averaged several hundred scans of the object, then ran our algorithm to segment the target and solve for R and T. Results are summarized in Table 1.

6 Conclusions and Future Work

We have presented a unique target object design and algorithm for automatic calibration of a 2D laser range sensor. Unlike previous methods, ours requires neither additional sensors such as cameras nor motion of the lidar, affording a great deal of flexibility and generality. We have derived target geometries that support estimation of extrinsic pose from a single cross sectional range measurement, and suggested a specific pyramid design based on a quadropod. Our techniques were demonstrated to exhibit robustness and accuracy, reliably locating the target amidst clutter and estimating pose to within less than 1° of rotation error and a few cm of position error for realistic sensor characteristics.

Our technique relies on precise construction of the calibration target, and requires that the scan's slice plane fall within a valid band that intersects all four faces of the pyramid. Because of finite angular resolution, the target must be placed within a small enough radius (empirically on the order of 20 meters) that it produces a sufficient number of samples on the target surface for reliable estimation. In practice, the target also must be held still so that range errors may be diminished via averaging of multiple scans.

We are currently working to incorporate this technique into a larger end-to-end system for self-consistent calibration of heterogeneous multi-modal sensor suites. For true absolute pose estimation – e.g. relative to an inertial frame rather than to the calibration target – a precise and repeatable method must be developed to measure the target's pose with respect to the frame of interest. Our pyramid target could be modified to simultaneously calibrate cameras by placing visible fiducials or patterns on its surface. Finally, it would be interesting to study the effect of relative target dimensions on pose estimation.

References

[1] Sick laser measurement systems technical description. Product Data Sheet, 2002.

[2] H. Andreasson, R. Triebel, and W. Burgard. Improving plane extraction from 3d data by fusing laser data and vision. In *IEEE IROS*, pages 2656–2661, August 2005.

[3] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[4] O. Jokinen. Self-calibration of a light striping system by matching multiple3-d profile maps. In *Second International Conference on 3-D Digital Imaging and Modeling*, pages 180–190, 1999.

[5] A. M. McIvor. Calibration of a laser stripe profiler. In *2nd International Conference on 3D Digital Imaging and Modeling (3DIM)*, pages 92–98, 1999.

[6] C. Mei and P. Rives. Calibration between a central catadioptric camera and a laser range finder for robotic applications. In *IEEE ICRA*, pages 532–537, May 2006.

[7] B. Tiddeman, M. Duffy, G. Rabey, and J. Lokier. Laser-video scanner calibration without the use of a frame store. *Vision, Image and Signal Processing*, 145(4):244–248, Aug 1998.

[8] G. Q. Wei and G. Hirzinger. Active self calibration of hand-mounted laser range finders. *IEEE Trans. Robotics and Automation*, 14(3):493–497, 1998.

[9] Q. Zhang and R. Pless. Constraints for heterogeneous sensor auto-calibration. In *IEEE Workshop on Realtime 3D Sensors and Their Use*, pages 38–43, 2004.

[10] Q. Zhang and R. Pless. Extrinsic calibration of a camera and laser range finder. In *IEEE IROS*, pages 2301–2306, 2004.

[11] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

A Pose from Conic Section

We briefly describe a technique for estimating pose from a conic cross section. Let $A = RK$ and $B = A^T SA$; then (6) becomes

$$q^T A^T SAq = q^T Bq = 0. \tag{11}$$

Note that the constraints are linear in the entries of matrix B. Since B is symmetric, and since it is defined only up to scale due to the homogeneity of (11), it has 5 degrees of freedom. A set of 5 linear constraints (i.e. 5 distinct points q) is therefore required for a unique solution, with each constraint of the form $c^T b = 0$, where

$$c^T = \begin{pmatrix} u^2 & 2uv & 2u & v^2 & 2v & 1 \end{pmatrix} \tag{12}$$

and b is a vector defining the relevant entries of the matrix B as

$$b^T = \begin{pmatrix} B_{11} & B_{12} & B_{13} & B_{22} & B_{23} & B_{33} \end{pmatrix}. \tag{13}$$

We form a constraint matrix C whose rows encode measured points (u_k, v_k) and take the form $c^T b = 0$, and solve the homogeneous system $Cb = 0$ by computing the eigenvector corresponding to the smallest eigenvalue of $C^T C$. We then factor the symmetric matrix B as $B = V\Lambda V^T$, where the columns of V are the eigenvectors and where Λ is a diagonal matrix of the eigenvalues. Using (11), we have

$$B = V\Lambda V^T = A^T SA, \tag{14}$$

so it follows that $A = \sqrt{S\Lambda}V^T$. The resulting matrix A may now be factored to obtain R and T as in Section 4.2. To refine these initial closed-form parameters, we again optimize using an iterative nonlinear algorithm; in particular, we estimate the pose parameters such that the sum of squared distances between RKq_k and the cone surface is minimized.

A Combined RANSAC-Hough Transform Algorithm for Fundamental Matrix Estimation

Richard J.M. den Hollander[†] Alan Hanjalic[‡]

[†] TNO Defence, Security and Safety
Oude Waalsdorperweg 63
P.O. Box 96864, 2509 JG Den Haag, The Netherlands
richard.denhollander@tno.nl

[‡] Information and Communication Theory Group
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
P.O. Box 5031, 2600 GA Delft, The Netherlands

Abstract

In this paper we will consider a combination of the RANSAC algorithm and the Hough transform for fast model estimation under the presence of outliers. The model will be computed by sampling a smaller than minimal subset, followed by a voting process of the remaining data. In order to use the combined method for this purpose, an adequate parameterization of the model in the Hough space is required. We will show that in case of hyperplane and fundamental matrix estimation, there is a similar and very general parameterization possible. It will allow these models to be estimated in a very efficient manner.

1 Introduction

The Hough transform determines for every data point the parameter subspace of models it supports, and increases the votes in the Hough space for all these models. An extension of this principle is to vote for sets of data points instead of single points. The subspace of supported models is then smaller, while the number of different point sets is larger. For example, a hyperplane in $\mathbb{R}^N$ is specified by N points, and a single point imposes a $N-1$ dimensional subspace of supported models in the Hough space. When a pair of points is considered, the set of supported models is a $N-2$ dimensional subspace. The voting process is then faster, but we have to consider $\binom{n}{2}$ different pairs instead of only n points. The limiting case is when precisely sets of N points are selected, which then results in a single point in the Hough space. This is the principle of the randomized Hough transform [11]. Instead of the total number of possible sets $\binom{n}{N}$, only a small number of random sets is selected which is sufficient to find the best model.

In contrast to the Hough transform, the RANSAC algorithm [3] samples N points and verifies the amount of support for the corresponding model. In view of the above, it is also possible to sample less than N points and verify the support for each supported model in the parameter subspace. This use of RANSAC in combination with the Hough

transform has been proposed in [8, 9] to improve the efficiency and quality of model estimation. It was argued that using sets of $N-1$ points is probably the best choice in terms of efficiency. This results in a one-dimensional subspace of models, which may be parameterized by a single quantity. Then there is no need to accumulate the large N dimensional Hough space.

The number of iterations J needed in the RANSAC algorithm is determined from the required probability of success, i.e. the probability that at least one all-inlier sample is found in J iterations [3]. Let ε denote the outlier ratio in the data, and d the number of points needed to hypothesize a model. If p is the probability of success, e.g. 0.99, then we have the relation

$$p = 1 - (1 - (1 - \varepsilon)^d)^J \tag{1}$$

The necessary number of iterations of the combined RANSAC and Hough method is clearly lower than for standard RANSAC, since only sets of $d = N-1$ instead of $d = N$ points are sampled for forming model hypotheses.

In general, an explicit parameterization of the fundamental matrix in the Hough space is impractical. Its estimation requires a 7-dimensional voting array (due to the 7 degrees of freedom [4]), which becomes unmanageable even for a moderate number of quantization levels. To be able to use the method in [8], we propose a new parameterization for hyperplanes which can also be applied to the fundamental matrix. The parameterization is based on the nullspace of a sample, where the sample will contain one point less than the minimally required number. For hyperplane estimation, we can include the threshold for the support set directly into the voting process. As a result, the whole range of models supported by the remaining data is taken into account. For fundamental matrix estimation, the correspondences will vote for single models. The resulting estimation by 6-point samples will be very efficient due to the reduced number of iterations. In [8] the quality of the model was also improved by using an error propagation mechanism for the data. Error propagation is not incorporated in our method, since no explicit parameterization of the model is used. Note that the standard RANSAC algorithm also neglects noise effects of points in the sample [2].

Several other modifications of RANSAC have been proposed to speed up the algorithm; the most directed to homography or fundamental matrix estimation. For example, in [10] the feature matching score is used in the selection probabilities of the correspondences in order to sample inliers more often. In [2], hypothesized models are optimized to compensate for noisy inliers and the resulting loss of support points. A faster support set evaluation has been proposed in [1], where a small number of randomly selected points is initially evaluated for support. Only when the hypothesized model has sufficient support points among this number, the remaining data is tested for support.

All these methods apply different speed-up mechanisms than our algorithm, and can therefore be combined with our algorithm to achieve even faster fundamental matrix estimation.

In Section 2 the proposed parameterization technique is discussed for hyperplane estimation. Section 3 describes the application of the method to fundamental matrix estimation. In Section 4, hyperplane and fundamental matrix estimation are evaluated on range data and real image pairs, respectively. Section 5 will conclude the paper.

2 Hyperplane estimation

The data points $\mathbf{x}_i$ for $i = 1,\ldots,n$ in $\mathbb{R}^N$ will be denoted by $\mathbf{x} = (x_1, x_2, \ldots, x_N)^\top$. A hyperplane with normal vector $\mathbf{n} = (a_1, a_2, \ldots, a_N)^\top$ and offset b is given by $a_1x_1 + a_2x_2 + \ldots + a_Nx_N + b = 0$. In short, the parameters of the hyperplane will be indicated by $\mathbf{h} = (\mathbf{n}^\top \; b)^\top$. The random samples that will be drawn consist of $N-1$ points $\{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \ldots, \tilde{\mathbf{x}}_{N-1}\}$, and solving for the hyperplane

$$\begin{bmatrix} \tilde{\mathbf{x}}_1^\top & 1 \\ \tilde{\mathbf{x}}_2^\top & 1 \\ \vdots & \vdots \\ \tilde{\mathbf{x}}_{N-1}^\top & 1 \end{bmatrix} \mathbf{h} = \mathbf{0} \tag{2}$$

yields a two-dimensional space $\{\mathbf{h_1}, \mathbf{h_2}\}$ for $\mathbf{h}$. This nullspace can in practice be computed by a singular value decomposition of the lefthand-side matrix. If the sample $\{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \ldots, \tilde{\mathbf{x}}_{N-1}\}$ contains only inliers, then the true hyperplane can be given by a linear combination of the nullspace vectors as

$$\mathbf{h} = \alpha\mathbf{h}_1 + (1-\alpha)\mathbf{h}_2 \tag{3}$$

The value of α can be found by solving $(\mathbf{x}^\top \; 1)\mathbf{h} = 0$ for another inlying point $\mathbf{x}$, and should be the same for all other inliers. The outliers will produce different values for α.

To find the true value of α we use a Hough-based voting mechanism for the remaining $n - N + 1$ data points [8]. We could use the projections of $\mathbf{x}$ onto $\mathbf{h}_1$ and $\mathbf{h}_2$ directly for computing α, but this may result in α values which are difficult to quantize. In particular, the nullspace vector with the largest singular value, say $\mathbf{h_1}$, is likely to constitute the largest part of $\mathbf{h}$ and therefore $\alpha \approx 1$. The binning of many values close to 1 and possibly some values far from 1 is impractical. It would be more convenient to have an α with equiprobable values over a large range.

For this purpose, we will make use of an orthonormal basis $\{\mathbf{u}_1, \mathbf{u}_2\}$ for the space spanned by $\mathbf{n}_1$ and $\mathbf{n}_2$, which are the normals in $\mathbf{h}_1$ and $\mathbf{h}_2$ from (3). We will take a point $\tilde{\mathbf{x}}_1$ from the sample, and project all vectors $\mathbf{x}_i - \tilde{\mathbf{x}}_1$ for $i = 1,\ldots,n$ (except those from the sample) onto this basis. The point $\tilde{\mathbf{x}}_1$ can be seen as the origin for the space spanned by $\{\mathbf{u}_1, \mathbf{u}_2\}$, which is shown in Fig. 1 for a line in 2D. From (3) we have that $\mathbf{n} = \alpha\mathbf{n}_1 + (1-\alpha)\mathbf{n}_2$, and since $\mathbf{n}_1$ and $\mathbf{n}_2$ are linear combinations of $\{\mathbf{u}_1, \mathbf{u}_2\}$ we can write

$$\mathbf{n} = c_1\mathbf{u}_1 + c_2\mathbf{u}_2 \tag{4}$$

for certain values c_1 and c_2. It then follows, that for the inliers the ratio of projections onto $\mathbf{u}_2$ and $\mathbf{u}_1$ becomes

$$\begin{aligned} \frac{(\mathbf{x}-\tilde{\mathbf{x}}_1)^\top\mathbf{u}_2}{(\mathbf{x}-\tilde{\mathbf{x}}_1)^\top\mathbf{u}_1} &= \frac{(\mathbf{x}-\tilde{\mathbf{x}}_1)^\top(\mathbf{n}-c_1\mathbf{u}_1)\frac{1}{c_2}}{(\mathbf{x}-\tilde{\mathbf{x}}_1)^\top\mathbf{u}_1} \\ &= \frac{(\mathbf{x}-\tilde{\mathbf{x}}_1)^\top\mathbf{u}_1\frac{-c_1}{c_2}}{(\mathbf{x}-\tilde{\mathbf{x}}_1)^\top\mathbf{u}_1} \\ &= \frac{-c_1}{c_2} \end{aligned} \tag{5}$$

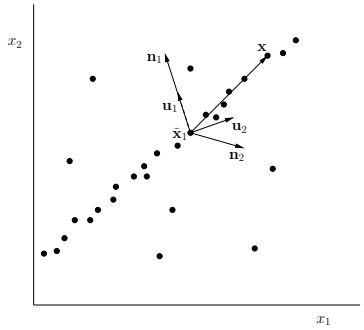


Figure 1: The sampled point $\tilde{\mathbf{x}}_1$ will serve as the origin for the space spanned by $\{\mathbf{n}_1, \mathbf{n}_2\}$. Each point $\mathbf{x}$ is projected onto this space by projecting the vector $\mathbf{x} - \tilde{\mathbf{x}}_1$ onto the orthonormal basis $\{\mathbf{u}_1, \mathbf{u}_2\}$.

since $(\mathbf{x} - \tilde{\mathbf{x}}_1)^\top \mathbf{n} = -b - (-b) = 0$. The outliers will produce different values for the projection ratio since in that case $\mathbf{x}^\top \mathbf{n} \neq -b$. The projection ratio in (5) will cover a relatively large range of values, and the angle γ of the projected vector $\mathbf{x} - \tilde{\mathbf{x}}_1$ with respect to the basis $\{\mathbf{u}_1, \mathbf{u}_2\}$

$$\gamma = \arctan\left(\frac{(\mathbf{x} - \tilde{\mathbf{x}}_1)^\top \mathbf{u}_2}{(\mathbf{x} - \tilde{\mathbf{x}}_1)^\top \mathbf{u}_1}\right) \tag{6}$$

offers a quantity which can conveniently be used in a voting space.

In principle, not only the hyperplane which crosses a point $\mathbf{x}$ should receive a vote, but all possible hyperplanes that are within allowable distance from the point. A data point will support a hyperplane if its orthogonal distance to the hyperplane is smaller than a threshold T (which is usually chosen heuristically in the RANSAC algorithm), see Fig. 2. Here the angle β determines for which models the indicated point can possibly be

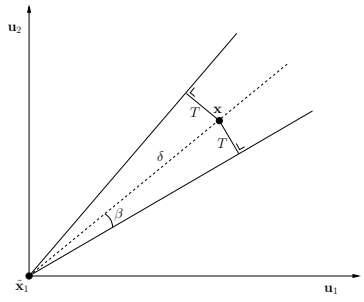


Figure 2: The projection of the point $\mathbf{x}$ in the frame $\{\mathbf{u}_1, \mathbf{u}_2\}$. There is a range of hyperplanes which the point supports. The maximum angle β for the range depends on threshold T and the length δ of the projection of the vector $\mathbf{x} - \tilde{\mathbf{x}}_1$.

a support point, and we have $\sin(\beta) = \frac{T}{\delta}$ where $\delta = \sqrt{((\mathbf{x} - \tilde{\mathbf{x}}_1)^\top \mathbf{u}_1)^2 + ((\mathbf{x} - \tilde{\mathbf{x}}_1)^\top \mathbf{u}_2)^2}$. A data point will vote for all angles in the range $[\gamma - \beta, \gamma + \beta]$. We note that the distance

from point $\mathbf{x}$ to the hyperplane is equal to the projected distance in the space spanned by $\{\mathbf{u}_1, \mathbf{u}_2\}$, since the component of $\mathbf{x}$ that lies outside this space is orthogonal to it.

The angle γ will be measured in degrees and we choose to use a voting space of 180 bins; one bin for each degree from -90 to 89. After calculating this angle for all points, it should result in a large number of votes in the bin of the true angle. A drawback of using all data points for voting, is that the voting operation may become quite complex for large data sets. Following the concept of the probabilistic Hough transform [6], we can also examine a subset of randomly sampled data points and calculate the best angle for this subset. This should give a sufficiently accurate estimate of the angle γ while making the voting process much faster. In the experiments we have chosen for a total of 100 randomly sampled data points, and only in case $n \leq 100$ we use all data.

The bin containing most votes determines the angle γ^* for which the final hyperplane is calculated according to

$$\mathbf{h} = \begin{pmatrix} \mathbf{u}_1 + \tan(\gamma^* + \frac{1}{2}\pi)\mathbf{u}_2 \\ -\tilde{\mathbf{x}}_1^\top(\mathbf{u}_1 + \tan(\gamma^* + \frac{1}{2}\pi)\mathbf{u}_2) \end{pmatrix} \tag{7}$$

where point $\tilde{\mathbf{x}}_1$ is taken from the sample.

3 Fundamental matrix estimation

The fundamental matrix can be estimated by following roughly the same technique as for hyperplane estimation. However, there are two major differences with the preceding scenario.

First, the 7-point algorithm uses the singularity constraint to determine the fundamental matrix. After seven correspondences are selected, solving for the fundamental matrix yields a two-dimensional nullspace [4]. Then the singularity constraint of the fundamental matrix needs to be used to find the solution. If we sample six points, the resulting nullspace is three-dimensional. We would like to use the singularity constraint for removing one dimension and use voting to find the final solution. Unfortunately, the singularity constraint on the three-dimensional nullspace is a cubic polynomial in two variables, which does not allow voting with respect to a fixed pair of nullspace vectors. As a result, we have to solve the singularity constraint for each correspondence individually. The complexity of the algorithm will therefore increase, but as we have already indicated, a subset of the data points will suffice in the voting process.

Second, since there is no fixed two-dimensional nullspace during voting, we can not calculate the range of allowable models as in Fig. 2. The number of fundamental matrices consistent with a seventh correspondence will be either one or three, just like for the 7-point algorithm. Therefore, there is no range of matrices for which e.g. the Sampson distance can be evaluated, and votes are cast for either one or three separate angles.

To start the estimation process we sample 6 correspondences $\{\tilde{\mathbf{x}}_1 \leftrightarrow \tilde{\mathbf{x}}'_1, \ldots, \tilde{\mathbf{x}}_6 \leftrightarrow \tilde{\mathbf{x}}'_6\}$, and solve

$$\begin{bmatrix} \tilde{x}'_1\tilde{x}_1 & \tilde{x}'_1\tilde{y}_1 & \tilde{x}'_1 & \tilde{y}'_1\tilde{x}_1 & \tilde{y}'_1\tilde{y}_1 & \tilde{y}'_1 & \tilde{x}_1 & \tilde{y}_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \tilde{x}'_6\tilde{x}_6 & \tilde{x}'_6\tilde{y}_6 & \tilde{x}'_6 & \tilde{y}'_6\tilde{x}_6 & \tilde{y}'_6\tilde{y}_6 & \tilde{y}'_6 & \tilde{x}_6 & \tilde{y}_6 & 1 \end{bmatrix} \mathbf{f} = \mathbf{0} \tag{8}$$

which results in a three-dimensional space of solutions

$$\mathbf{f} = \alpha\mathbf{f}_1 + \beta\mathbf{f}_2 + (1-\alpha-\beta)\mathbf{f}_3 \tag{9}$$

If we take a single correspondence $\mathbf{x} \leftrightarrow \mathbf{x}'$ and solve

$$\begin{pmatrix} x'x & x'y & x' & y'x & y'y & y' & x & y & 1 \end{pmatrix}\mathbf{f} = 0 \tag{10}$$

for $\mathbf{f}$ from (9) we get a linear constraint in α and β. When the correspondence is an inlier, the true values for α and β will satisfy this constraint. Let the resulting linear relation be $\beta = r\alpha + g$. Then we use the singularity constraint

$$\det(\alpha F_1 + (r\alpha+g)F_2 + (1-\alpha-(r\alpha+g))F_3) = 0 \tag{11}$$

where F_1, F_2, and F_3 are the 3×3 matrices containing the elements of $\mathbf{f}_1$, $\mathbf{f}_2$ and $\mathbf{f}_3$, respectively. This will result in either one or three real solutions for α and thus for $\mathbf{f}$. Now, writing the vectors $\mathbf{f}_1$, $\mathbf{f}_2$ and $\mathbf{f}_3$ in (9) as $\mathbf{f}_1 = (\mathbf{n}_1^\top\ b_1)^\top$, $\mathbf{f}_2 = (\mathbf{n}_2^\top\ b_2)^\top$ and $\mathbf{f}_3 = (\mathbf{n}_3^\top\ b_3)^\top$, we construct an orthonormal basis $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$ from $\{\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3\}$. This basis is used for the projection of the solutions for $\mathbf{f}$. In particular, we calculate the angles

$$\gamma_1 = \arctan\left(\frac{(f_1\cdots f_8)\mathbf{u}_2}{(f_1\cdots f_8)\mathbf{u}_1}\right) \qquad \gamma_2 = \arctan\left(\frac{(f_1\cdots f_8)\mathbf{u}_3}{(f_1\cdots f_8)\mathbf{u}_2}\right) \tag{12}$$

and use them to cast a vote in a two-dimensional array. The angles will be rounded towards full degrees in the range -90 to 89.

As in hyperplane estimation, we do not use all data points during voting. When the data set contains more than 100 correspondences, only 100 randomly selected correspondences are considered. Examples of vote distributions are given in [5].

After having located the values of γ_1^* and γ_2^* for the bin containing most votes, we can find the first eight elements of the corresponding $\mathbf{f}$ by

$$\begin{pmatrix} f_1 \\ \vdots \\ f_8 \end{pmatrix} = \mathbf{u}_1 + \tan(\gamma_1^*)\mathbf{u}_2 + \tan(\gamma_1^*)\tan(\gamma_2^*)\mathbf{u}_3 \tag{13}$$

and the last element by

$$f_9 = -\begin{pmatrix} \tilde{x}_1'\tilde{x}_1 & \tilde{x}_1'\tilde{y}_1 & \tilde{x}_1' & \tilde{y}_1'\tilde{x}_1 & \tilde{y}_1'\tilde{y}_1 & \tilde{y}_1' & \tilde{x}_1 & \tilde{y}_1 \end{pmatrix}\begin{pmatrix} f_1 \\ \vdots \\ f_8 \end{pmatrix} \tag{14}$$

The correspondence $\tilde{\mathbf{x}}_1 \leftrightarrow \tilde{\mathbf{x}}_1'$ is part of the 6-point sample, and therefore lies on the final $\mathbf{f}$.

The fundamental matrix that is found this way does not automatically satisfy the singularity constraint. Due to the rounding effect in the voting array, the matrix will slightly deviate from a singular one. We can solve this by applying the SVD to this matrix, and setting the smallest singular value to zero [4]. A prerequisite for this to work properly is a normalization of the correspondences. This entails a translation which results in zero means for the (x,y) coordinates, followed by a scaling which makes their average distance to the origin equal to $\sqrt{2}$. The transformation is applied to both images' correspondences

independently. Before the support of the fundamental matrix is evaluated, the coordinates are transformed back again to their original values.

The whole sequence of steps in the estimation process is listed in Fig. 3. Note that the number of iterations J is determined adaptively as in [4]. When the largest support set so far is found, i.e. $|S_j| > |S_{max}|$, the outlier ratio ε is updated accordingly. The number of iterations J is then recomputed according to (1).

- $j = 1, \quad J = \infty, \quad S_{max} = \emptyset$
- Normalize the image correspondences.
- **while** $j < J$ **do**
  - Randomly select 6 correspondences $\{\tilde{\mathbf{x}}_1 \leftrightarrow \tilde{\mathbf{x}}'_1, \ldots, \tilde{\mathbf{x}}_6 \leftrightarrow \tilde{\mathbf{x}}'_6\}$ and use them to compute their nullspace $\{\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3\}$ by solving (8).
  - Determine the orthonormal basis $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$ for the space spanned by the normals in $\{\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3\}$.
  - **if** $n > 100$ **then**
    - form the set C by randomly selecting 100 correspondences from $\{\mathbf{x}_1 \leftrightarrow \mathbf{x}'_1, \ldots, \mathbf{x}_n \leftrightarrow \mathbf{x}'_n\} \setminus \{\tilde{\mathbf{x}}_1 \leftrightarrow \tilde{\mathbf{x}}'_1, \ldots, \tilde{\mathbf{x}}_6 \leftrightarrow \tilde{\mathbf{x}}'_6\}$
  - **else**
    - form $C = \{\mathbf{x}_1 \leftrightarrow \mathbf{x}'_1, \ldots, \mathbf{x}_n \leftrightarrow \mathbf{x}'_n\} \setminus \{\tilde{\mathbf{x}}_1 \leftrightarrow \tilde{\mathbf{x}}'_1, \ldots, \tilde{\mathbf{x}}_6 \leftrightarrow \tilde{\mathbf{x}}'_6\}$
  - **for each** $\mathbf{x} \leftrightarrow \mathbf{x}'$ **in** C **do**
    - Find the possible solutions for $\mathbf{x} \leftrightarrow \mathbf{x}'$ by solving (10) and (11).
    - Determine γ_1 and γ_2 according to (12) for each solution, and round the angles to the nearest degree.
    - Add one vote for each pair of angles (γ_1, γ_2) in the voting array.
  - Determine the pair (γ_1^*, γ_2^*) with the maximum number of votes.
  - Construct $\mathbf{f}$ from (13) and (14) for γ_1^* and γ_2^*.
  - Find the closest approximation $\hat{\mathbf{f}}$ to $\mathbf{f}$ with $\det(\hat{F}) = 0$ using the SVD.
  - Determine the set of support points S_j for the denormalized $\hat{\mathbf{f}}$, by verifying which points are within distance T.
  - **if** $|S_j| > |S_{max}|$ **then**
    - $J = \log(1-p) \cdot \log^{-1}\left(1 - \left(\frac{|S_j|}{n}\right)^6\right)$
    - $S_{max} = S_j$
  - $j = j + 1$
- Re-estimate the fundamental matrix based on the largest support set S_{max}.

Figure 3: The RANSAC-Hough algorithm for fundamental matrix estimation using a two-dimensional voting space.

	$\sum J_R$ $(\cdot 10^3)$	$\sum J_{RH}$ $(\cdot 10^3)$	t_R	t_{RH}	$\sum \lvert S_{max}\rvert_R$ $(\cdot 10^4)$	$\sum \lvert S_{max}\rvert_{RH}$ $(\cdot 10^4)$	$\#\mathbf{h}_R$	$\#\mathbf{h}_{RH}$
image 0	9.50 ± 1.42	1.63 ± 0.17	19.9 ± 2.37	3.84 ± 0.31	6.29 ± 0.032	6.29 ± 0.033	14.0 ± 0.7	13.8 ± 0.6
image 1	3.41 ± 0.48	0.72 ± 0.09	5.47 ± 0.64	1.39 ± 0.13	6.37 ± 0.029	6.37 ± 0.028	10.8 ± 0.6	10.4 ± 0.6
image 2	5.36 ± 0.75	1.05 ± 0.11	9.58 ± 1.53	2.06 ± 0.24	6.41 ± 0.034	6.41 ± 0.034	13.0 ± 0.7	12.7 ± 0.6
image 3	2.12 ± 0.37	0.51 ± 0.07	4.13 ± 0.70	1.13 ± 0.14	6.41 ± 0.028	6.41 ± 0.026	10.0 ± 0.6	9.7 ± 0.6
image 4	2.88 ± 0.50	0.63 ± 0.08	5.71 ± 0.73	1.45 ± 0.14	6.41 ± 0.029	6.40 ± 0.028	11.1 ± 0.7	10.6 ± 0.6
image 5	2.55 ± 0.34	0.58 ± 0.07	4.32 ± 0.57	1.14 ± 0.13	6.39 ± 0.029	6.39 ± 0.030	10.0 ± 0.6	9.7 ± 0.6
image 6	5.03 ± 0.68	1.00 ± 0.11	9.80 ± 1.18	2.18 ± 0.20	6.36 ± 0.029	6.36 ± 0.031	12.3 ± 0.6	12.0 ± 0.6
image 7	1.62 ± 0.29	0.41 ± 0.05	3.74 ± 0.39	1.07 ± 0.08	6.44 ± 0.026	6.44 ± 0.024	9.7 ± 0.6	9.3 ± 0.5
image 8	2.00 ± 0.36	0.47 ± 0.07	3.61 ± 0.57	1.00 ± 0.11	6.40 ± 0.028	6.40 ± 0.028	9.2 ± 0.6	8.9 ± 0.6
image 9	1.91 ± 0.37	0.46 ± 0.07	4.09 ± 0.52	1.14 ± 0.10	6.44 ± 0.029	6.44 ± 0.029	10.2 ± 0.7	9.7 ± 0.7

Table 1: The results for finding all planes using RANSAC (R) and RANSAC-Hough (RH) in the ABW range images. Indicated are the averages and standard deviations ($\pm$) for the total number of iterations $\sum J$ per image, the running time t in seconds, the total size of the maximum support sets $\sum |S_{max}|$ and the number of planes $\#\mathbf{h}$ found.

4 Experimental results

We will compare the proposed RANSAC-Hough method with the standard RANSAC algorithm for plane fitting and fundamental matrix estimation. For all experiments we report the averages and standard deviations over a number of runs of both the executed number of iterations J and the size of the maximum support set $|S_{max}|$. Furthermore, the average running time for a single run is listed. The final re-estimation step in RANSAC will be omitted. The algorithms were implemented in C and ran on Intel Xeon 3.07 GHz / 3.2 GHz computers. For implementation details see [5].

4.1 Plane fitting

As application we consider the fitting of planes in range image data. We have used 10 images ("train" 0 to 9) from the ABW structured light scanner in the USF database[1]. The images contain several different planar objects, and the intensity values correspond to the measured depth by the scanner. An example of one of the images is shown in Fig. 4(a). We have subsampled the images with a factor 2 to obtain 256×256 sized images. We search with RANSAC for planes in the images, and subsequently delete the points from the data set which belong to a plane. The repeated application of RANSAC is stopped when a plane is returned with support smaller than 500 pixels. For the shown example image, the number of planes extracted this way will be about 12. The experiment is repeated 500 times for each image. The threshold for the orthogonal distance to the plane is set to $T = 2.5$, which is large enough to capture noisy variations of the inliers. Table 1 shows the results of the experiments.

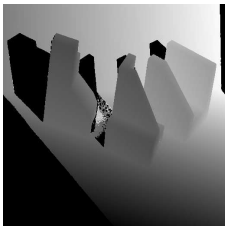
The RANSAC-Hough method outperforms RANSAC in all aspects; in some cases it is up to a factor 5 faster. The total number of points on the extracted planes is comparable, while the number of planes is a bit smaller. This means that the extracted planes are actually better fits, since they contain a larger part of the data.

4.2 Fundamental matrix estimation

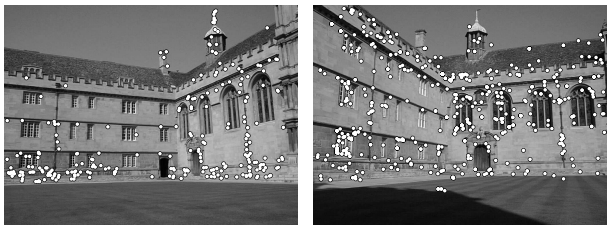
Some of the real images we have used for testing are shown in Fig. 4(b) and 4(c). There are differences in viewpoint and/or zoom factor between the left and right images. The

[1] Available at http://marathon.csee.usf.edu/range/seg-comp/images.html.

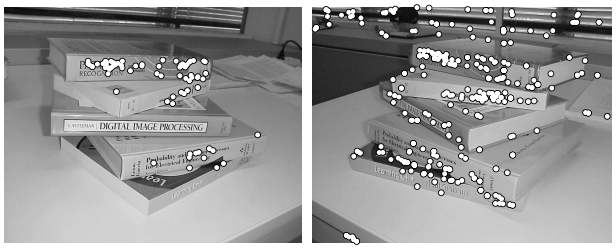
SIFT keypoint detector[2] [7] has been applied for establishing correspondences between the image pairs. The left images in Fig. 4(b) to 4(c) show the final sets of inlying feature points, and the right images the outlying feature points. Also indicated for every image pair are the total number of correspondences n and the outlier ratio ε. The Sampson distance is chosen as error measure, and we have set the square root of the threshold to $T = 1.5$ pixels.



(a) A range image used for plane fitting.



(b) Wadham college: $n = 921$ and $\varepsilon = 0.71$.



(c) Pile of books: $n = 548$ and $\varepsilon = 0.82$.

Figure 4: Some of the images used in the experiments.

The results of running RANSAC 500 times on the image pairs are shown in Table 2. The difference in running times is best noticeable for higher outlier ratios. The number of iterations is reduced here considerably and the additional complexity of the voting process does not prohibit a speedup anymore. The support sets found are slightly smaller than those for RANSAC. This is a result of the rank 2 enforcement which finds an approximation of the fundamental matrix computed from the data. Since the data is not considered in finding this approximation, some inliers are lost in the process.

5 Discussion

The combination of RANSAC and the Hough transform, that has been advocated in the past, is made applicable to hyperplanes and the fundamental matrix by a new parameterization of the model. For hyperplanes, the result is an efficient one-dimensional voting space and a reduction of the sample size by one point. For the fundamental matrix, a two-dimensional voting space is applied because of the singularity constraint. Instead of sampling 7 correspondences per model, we now only need to take 6-point samples. This

[2] The code is obtained from http://www.cs.ubc.ca/∼lowe/keypoints/.

image pair	ε	# inliers	J_R	J_{RH}	$\lvert S_{max}\rvert_R$	$\lvert S_{max}\rvert_{RH}$	t_R	t_{RH}
books	0.74	189	$7.2 \pm 0.64 \cdot 10^4$	$1.92 \pm 0.17 \cdot 10^4$	187 ± 2.5	185 ± 2.9	25.1 ± 3.6	11.7 ± 1.9
pile of books	0.82	97	$4.03 \pm 0.71 \cdot 10^5$	$0.93 \pm 0.18 \cdot 10^5$	109 ± 2.9	106 ± 3.5	114 ± 24.2	52.5 ± 12.0
Wadham college	0.71	264	$7.08 \pm 2.93 \cdot 10^4$	$1.96 \pm 0.73 \cdot 10^4$	241 ± 13.8	236 ± 14.4	29.0 ± 12.4	12.4 ± 4.9
Univ. British Columbia	0.56	399	$2.65 \pm 0.56 \cdot 10^3$	$1.14 \pm 0.25 \cdot 10^3$	372 ± 11.0	369 ± 12.8	1.13 ± 0.28	0.72 ± 0.20
Corridor	0.43	150	466 ± 160	261 ± 95.5	139 ± 5.8	138 ± 6.8	0.08 ± 0.031	0.13 ± 0.055
Valbonne church	0.58	127	$2.56 \pm 0.63 \cdot 10^3$	$1.23 \pm 0.40 \cdot 10^3$	123 ± 3.7	121 ± 5.4	0.49 ± 0.13	0.64 ± 0.23

Table 2: Fundamental matrix estimation using RANSAC (R) and RANSAC-Hough (RH) on real image pairs. Indicated are the averages and standard deviations ($\pm$) for the executed number of iterations J, the maximum number of support points $|S_{max}|$ and the running time t in seconds.

makes it much easier to find an all-inlier sample by random trials. In addition, we use for both models randomly selected subsets of the data to speed up the voting stage.

The consecutive extraction of planes in range images took considerably less time using the RANSAC-Hough method. The quality of the solutions is either equal or better than standard RANSAC. In case of the fundamental matrix, a much faster estimation is achieved for high outlier ratios, with only a minor decrease in the size of the support.

A further improvement of the algorithm may be circumventing the loss of support points caused by enforcement of the singularity constraint.

References

[1] O. Chum and J. Matas. Randomized ransac with $T_{d,d}$ test. In *Proc. British Machine Vision Conference*, 2002.

[2] O. Chum, J. Matas, and J. Kittler. Locally optimized ransac. In *Proc. DAGM*. Springer-Verlag, 2003.

[3] 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.

[4] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2003.

[5] R.J.M. den Hollander and A. Hanjalic. A six-point RANSAC algorithm for fundamental matrix estimation. Technical Report ICT-2007-02, Delft University of Technology, 2007.

[6] N. Kiryati, Y. Eldar, and A.M. Bruckstein. A probabilistic Hough transform. *Pattern Recognition*, 24(4):303–316, 1991.

[7] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, 60(2):91–110, 2004.

[8] C.F. Olson. Constrained Hough transforms for curve detection. *Computer Vision and Image Understanding*, 73(3), 1999.

[9] C.F. Olson. A general method for geometric feature matching and model extraction. *Int. Journal of Computer Vision*, 45(1), 2001.

[10] B. Tordoff and D.W. Murray. Guided sampling and consensus for motion estimation. In *Proc. European Conference on Computer Vision*, 2002.

[11] L. Xu, E. Oja, and P. Kultanen. A new curve detection method: Randomized Hough Transform (RHT). *Pattern Recognition Letters*, 11(5), 1990.

Automatic Identification of Morphometric Landmarks in Digital Images

Sasirekha Palaniswamy[1], Neil A Thacker[2] & Christian Peter Klingenberg[1]
[1]Faculty of Life Sciences, [2]Imaging Science and Biomedical Engineering,
The University of Manchester

Abstract

Our aim is to develop a completely automated and reliable system to identify morphological landmarks in digital images. The performance of the system is aimed to replicate manual digitization with equivalent accuracy and reliability, based upon a small number of training examples. The analysis system is constructed from four stages; a feature based detection of fly wing structure, correspondence matching based upon the pairwise geometric histogram (PGH) representation, global location of the wing using a Probabilistic Hough Transform (PHT), and finally local correlation based refinement of individual features. We evaluate this system and compare quantitative results to manually digitized data.

1 Background

Morphological landmarks are points that can be located precisely and establish an unambiguous one-to-one correspondence among all the specimens and are widely used in shape analysis [1]. Points like the tip of the nose or the outer corner of the left eye are possible landmark points of the human face. Analyses of shape investigate the arrangement of landmark points relative to each other. A substantial body of statistical methods is available for the analysis of configurations of landmark points [2].

This framework of shape analysis by landmarks is increasingly used in many biological and medical applications and widely applied in many other fields. The configuration of landmarks have helped identify the possible source of re-infesting specimens and encounter the epidemiologically challenging vectors of Chagas disease [3]. The potential of using geometrical morphometric techniques as an invaluable tool for recognizing taxonomic data is being explored [4]. Other scientific applications include investigating the study of size and shape to examine the effects of experimental treatments, genotype or other factors directly in the anatomical aspect. The use of landmarks has been adapted to specific biological contexts such as genetics [5, 6, 7]; geographic differentiation [8], and the study of morphological integration [9, 10].

The process of identifying the landmarks is an important and labour-intensive part of any such analysis. Presently, this is usually done manually. Plugins for the ImageJ software (for digitizing the standard sets of landmarks) on fly wings and mouse mandibles), increases speed and reliability over a completely unaided process [11]. However, there is still a requirement for an observer to manually identify each landmark point and therefore this process can be time-consuming, and quite often, the research questions are dependant on the duration of obtaining these data.

Developing an automated system for locating landmarks in digital images of *Drosophila* wings is largely significant, as it is an excellent model for the study of genetics of development and evolution of morphological form [12, 13]. They contain a wealth of interesting biological information and its simple, flat, two dimensional structure enables convenient handling. Therefore, automation has numerous advantages over a manual system, as it will not only diminish the labour needed for shape analysis, but it also will eliminate the source of error (mistakes made during digitizing and subtle differences between observers). Furthermore, automatic extraction of features from images can potentially change the way in which landmarks are chosen for morphometric studies. Whereas the traditional approach requires landmarks to be chosen a priori, based on outside knowledge of the study system, the approach using automated image analysis raises the possibility to identify and extract features from the total information contained in the images that are maximally informative in the context of a particular research project.

There have been previous attempts to automate the process of landmark location on the *Drosophila* wing [13]. This method is semi-automated where the operator initiates the process by marking two landmark positions and the system fits a series of spline curves to the margin and the veins of the wing, and defines the landmarks as the intersections of the splines. The drawback of this system is that the landmarks are not always at the exact location of the intersection of veins and the splines may not exactly match the veins (because of the "stiffness" of the spline interpolation) and the system has problems identifying wings of species with highly melanized spots at the intersections.

Another such system to locate the landmarks on digital images of bee wings is being developed at the Paris National Museum of Natural History [14]. This system applies the techniques of mathematical morphology and skeletonization to obtain the landmarks. However, using these techniques are not guaranteed to be robust. The method also requires human intervention in terms of loading the data and identifying the initial set of points to start the process and the pre-processing step includes certain parameters that have to be set by the operator.

Other similar automated systems include the Digital Automated Identification SYstem (DAISY) which was developed whilst attempting a novel approach to identify insect specimens from the images. Principal component auto-associative memories and trainable classifiers are exploited to identify closely related parasitic wasps based on their wing venation and pigmentation patterns [15]. This system has been designed to identify several organismal groups in real time and it successfully classifies data into morphologically similar classes and proves to be a very useful and practical tool for taxonomic identification of various species. The internal algorithms are based upon the use of a pairwise geometric histograms (PGH) representation,which is used to establish shape correspondence.

2 Introduction

We present an automated system for the analysis of edge based structure for use in morphometric studies. The current work takes a grey level image of *Drosophila* wing as input and extracts the coordinates of 15 landmarks (Figure 1). A typical shape analysis requires several hundred images and identifying these landmarks is a laborious process. An automated method to extract these features can potentially improve the methodology with

which the landmarks are identified via improved standardization and accuracy.

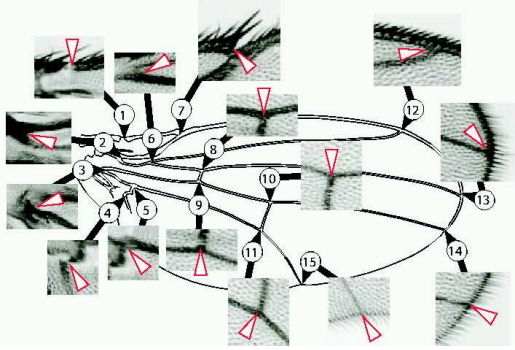


Figure 1: 15 Landmark locations on *Drosophila* wing (Image courtesy: [11]).

The proposed method extracts the ridges (linear features such as wing veins) using the knowledge of their known grey level profile and the noise characteristics of the image. This approach has been shown to be statistically valid [16]. The ridges obtained are approximated by line segments and the geometric relationships between them are encoded in PGH, an approximation to the probabilistic density function for the geometric co-occurrences in the data [17]. Shape correspondence is determined by comparing and matching the pairwise histograms of the scene and model data. A probabilistic Hough transform (robust Likelihood) is used to determine the hypothesized landmark location (Figure 2). Sub-pixel estimation of the landmark location is performed by template matching, i.e., correlating a small region around the Hough estimated landmark location.

We show that a single training image with its landmark coordinates is enough to independently estimate the landmarks of any individual within a particular dataset. However, the reliability and accuracy of the method can be further enhanced by using multiple training images. Multiple estimates also offer the possibility of accuracy assessment, an important aspect of any scientific study. The precision, repeatability and robustness of the algorithm have been evaluated here as a pilot study. Although some predictions regarding reliability can be made with a small sample, a further study will be carried out on a larger sample to test the reliability of the system on scientific studies.

3 Methods

3.1 Data Acquisition

Data acquisition is carried out by mounting the fly wings in rows on a microscopic slide and flattening with a coverslip. The digital images are obtained using an appropriate digital camera mounted on the microscope and attached to a computer. A calibration image is generally obtained along with each of the dataset to standardize the difference in magnifications between different dataset. The anatomical landmarks can be easily collected in two dimensions from digital images and this approach is quite useful in evolutionary

research as the landmarks can be collected from non-model organisms or even fossils. The X & Y co-ordinates of these landmarks are usually obtained by manually digitizing the location of these co-ordinates appropriately based on their anatomical context. Specialized algorithms and plugins can be used to semi-automate the process and to enhance the speed of the digitization.

3.2 Analysis

The analysis system is constructed from four stages; a feature-based detection of fly wing structure, correspondence matching based upon the PGH representation, global location of the wing using a Probabilistic Hough transform, and finally correlation based refinement of individual features. We evaluate this system and compare quantitative results to manually digitized data below.

3.2.1 Ridge Detection- An extension to Canny framework

The wing veins are extracted as ridge features, using a method which is a modification of more conventional edge detectors. We locate these features using a matched filter approach, approximating the vein profile as a Difference of Gaussians. Local maxima in response are then passed into a more conventional hysteresis threshold and linking system, based upon the popular Canny [18] system, in order to extract connected structures. This extracted edge map can be used to determine the precise location of landmarks. The uniformity of noise in the feature enhancement stage guarantees that this process is inherently stable. The ridge detector has been optimized for the task of locating landmarks by analyzing the specific characteristics of noise and scale stability. The whole process can be interpreted as a statistical null-hypothesis test for the presence of the defined feature [19].

3.2.2 Pairwise Geometric Histograms

The extracted edge-map is approximated by line segments and the geometric relationships between each pair of line segments are encoded in the pairwise geometric histograms. This is an approximation to the probabilistic density function,

$$H_i(\theta, d) \;=\; P(\theta_i - \theta_j, d_{ij} | e_i) \tag{1}$$

for the geometric co-occurences of an edgel e_j given e_i as a function of relative angle $\theta_i - \theta_j$ and perpendicular separation d_{ij}. This is a well established method of shape representation based on recording the distribution of pairwise geometric relationships between local shape features which can support recognition and there is considerable robustness to the loss of data due to fragmentation noise and occlusion [17]. The method is also known to be complete, in that the original structure of the object can be reconstructed from the set of histograms describing a shape. This representation is invariant for portions of the same linear feature so that it can be constructed by considering a linearization of the edge map. The importance of a pair of line segments defining the representative shape can be encoded by entering the product of their lengths at the value of the entry. The entry is blurred along each axis to encode the uncertainty regarding the true position and orientation of each line segment. The scale of binning and extent of blurring defines the extent of allowable differences when matching similar shapes.

Shape recognition is done by identifying the correspondences between image and object features. Shape representation comprises many geometric histograms, each representing a single model feature. The degree to which a linear edge feature in the test image matches a particular model feature can be determined by comparing their histograms. The degree of match between them is given by the Bhattacharrya measure B_{ij}, which takes the form of a dot product correlation of the histograms of lines H_i and H_j.

$$B_{ij} = \sum_{\theta}^{2\pi} \sum_{d}^{d_{max}} \sqrt{H_i(\theta,d) H_j(\theta,d)} \tag{2}$$

This can be related, via the χ^2 variable, to a maximum likelihood similarity metric and can be derived as an approximation to Fisher's Exact test as a method for comparing two distributions. The hypothesized matches can then be used as input to pose an estimation algorithm such as the generalized Hough transforms. Scale independent recognition can be achieved by representing an object at a range of scales [20]. However, this property of the PGH representation was not required in our study and therefore was not utilized.

3.2.3 Hough Transform

A probabilistic Hough Transform (robust maximum Likelihood), is used to make an estimate of the global position and orientation of each wing. Entries in the 2D location histogram are made according to the localization covariance, propagated from the errors on the constraint lines. This takes proper account of errors, resulting in improved robustness and more accurate determination of model position, orientation and scale in comparison to the more conventional form of this algorithm. The entries in the Hough arrays are constructed from pair of lines (n,m), i.e., a tuple transform. The equivalent probabilistic form for the Hough transform $H(x,y)$ used to find the position of a model in a scene is given by the expression,

$$H(x,y) = \sum_{n}^{N} \sum_{m}^{N} log(p(x,y|n)p(x,y|m)) = \sum_{n}^{N} log(p(x,y|n)) \sum_{m}^{N} log(p(x,y|m)) \tag{3}$$

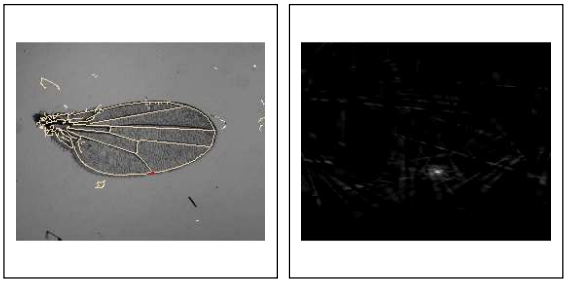
so that the Hough entry can be considered as the square of the robust log Likelihood $L(x,y)$ for the localization of the object,

$$H(x,y) = L(x,y)^2 = (\sum_{n}^{N} log(p(x,y|n)))^2 \tag{4}$$

During array construction $H_{nm} = log(p(x,y|n)p(x,y|m))$ is estimated from a 2D Gaussian distribution centered at the the position of the model hypothesized by the m,nth pair of scene line labels with variance propagated from the individual line location errors. This tuple-based construction helps to remove background noise from the Hough array and has some computational advantages. The variability of the line segmentation process and the uniform error on the scale estimates are independent and are adjusted to give a quantitative estimate of the hypothesized location of a pre-defined reference point from pair of scene lines. Training from example data involves recording the perpendicular distance, *'d'*, from each model line to the reference point. Consequently, for each pair

of scene lines, extended lines at the appropriate perpendicular distance will intersect at the hypothesized position of the reference. Error at the point of intersection can again be estimated by standard error propagation.

Models can be located based on the positions, orientations and scales hypothesized by scene line labels. However, the orientations and scales of the models are not determined explicitly. This can be determined separately using 1-parameter Hough transforms. For each model position determined, a 1-parameter orientation Hough transform and 1-parameter scale Hough transform can be constructed from entries selected on the basis of consistency between the scene lines and model position. The orientation is determined from the difference in orientation between the scene line and model line to which it matched. Comparing the perpendicular distance from the scene line to the model position to this same distance in the model itself would yield the scale. Peaks in these Hough transforms would give the orientation and scale of the model at that position in the scene.



(a) Flywing image with model overlaid. (b) Peak in the Hough transform.

Figure 2: Hough transform located 15th landmark.

3.2.4 Template Matching

The above Hough scheme computes an estimate of landmark position based upon global wing shape. As we need to determine variations in shape for the morphometric study this estimate needs to be refined based upon local image evidence. To obtain this estimate, template matching is performed on the Difference of Gaussian image of the scene $D(I)$ and model (example mark-up) $D(M)$ data, over a small region around the Hough estimate for the feature in the scene data. To save processing time during alignment, the scene data is rotated to match the model data using the Hough estimate, which is assumed to be sufficiently accurate for final location of the landmark. The use of the Difference of Gaussian images eliminates any image illumination offset and the matching is performed as a dot-product correlation in order to eliminate the effects of illumination scaling,

$$L_{h_x h_y} = \sum_{x}^{R}\sum_{y}^{R} D(M(x,y))D(I(x+h_x,y+h_y))/\sqrt{\sum_{x}^{R}\sum_{y}^{R} D(I(x+h_x,y+h_y))^2} \quad (5)$$

where, R is the region size. This is directly equivalent to performing a least squares comparison of the image regions with one free grey level scale parameter. In this study,

the denominator is presumed to be constant to save time on the computationally expensive calculations. The best possible match is identified and that location is transformed back onto the scene image. The least-squares difference between the two scale image regions is stored so that the best matching examples can be selected for final estimation of landmark position (see below). This quality control feature not only allows a check on the adequacy of the example mark-ups but also eliminates residual problems in alignment estimation, such as poor rotation estimates.

4 Results

4.1 Precision of manual digitization

The precision of manual digitization by an expert is determined by determining the deviation (difference between the value of each attempt and the mean) of 10 repeats of digitization of a single image. The outcome shows that it is within a range of +/- 1 pixel (Figure: 3).

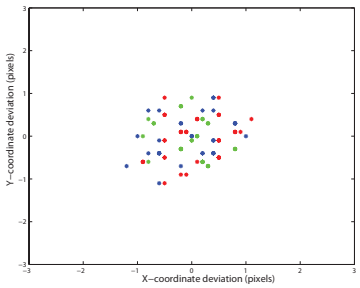
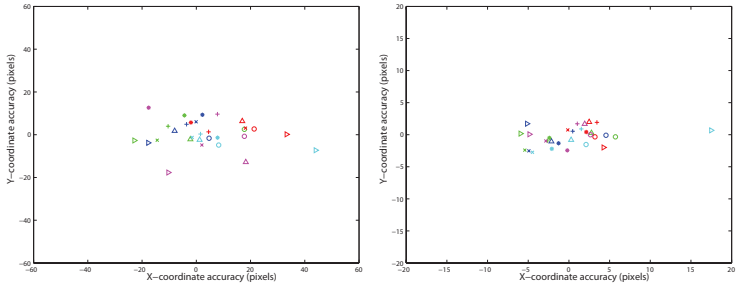


Figure 3: Reproducibility of landmark location over 10 repeats-manual digitization.

4.2 Accuracy of the template matching relative to manual digitization

To test the utility of the template matching stage, the feature (landmark 12) that was significantly variant in comparison to other landmarks was taken. Figure 4(a) shows the positions of the landmarks located by the Hough transform relative to the landmark locations digitized manually. The Hough transform locates most of the landmarks within a range of +/-20 pixels. This provides an estimate of the range that the correlation search must operate over, and is used to define a ‘window size’parameter. During initial testing, by ensuring that the window size is large enough, we can be certain that the landmarks can be located reliably. The refinement by the template matching strategy is shown in Figure 4(b) indicating the improvement in accuracy to be within a range of +/-6 pixels on the X-axis and +/-3 pixels on the Y-axis.



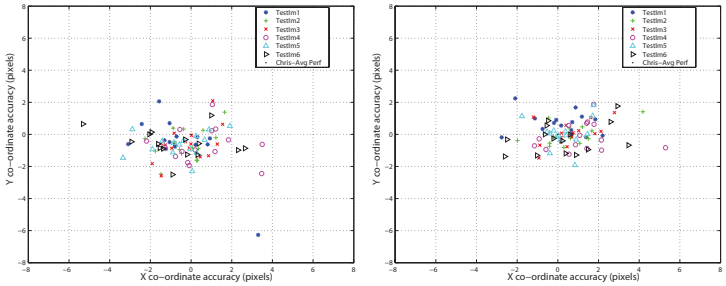
(a) Hough estimation vs actual location of landmark 12.

(b) Template matching refinement vs actual location of landmark 12.

Figure 4: Hough transform & template matching performance.

4.3 Accuracy of the automated system

The accuracy of the system is assessed using multiple reference images (multiple models). The results show that the landmarks can be located more precisely in cases where the model features are a good match to the scene data. However, it is quite unlikely that the model chosen would be suitable for all the features to be estimated. Therefore, it is important to choose a set of appropriate model features in the training data that can best match with the given test dataset. This can be achieved by computing the degree of least-squares match between the model and the test feature and taking an average of the best matches available. Since, the precise location of certain features (eg., landmark 12 due to its structural complexity) can be quite challenging in contrast to most other features, the degree of least-squares match can also be applied as a quality control approach to determine the adequacy of the selected training examples.



(a) Avg of best 3 least squares match(5 reference images)

(b) Avg of best 5 least squares match (11 reference images)

Figure 5: Best Least Squares Match.

The Figures 5(a) & 5(b) shows the system performance with an average of the best 3 of 5 reference images and best 5 of 11 reference images using the least squares match. It can be seen that the accuracy of the system has improved with an increase in the number of reference images. Most of the landmarks in the sample dataset have been located within +/-3 pixels accuracy. It should be noted that the outliers are mainly contributed by one of

the test image which clearly indicates that the reference images used were not necessarily a good match for feature location in that particular image. In such cases, increasing the number of reference images would considerably improve the accuracy.

4.4 Robustness of the system

The robustness of the system was tested by locating landmarks in an image with additive noise of $\tilde{10}$ times that of the original image. The system is robust in locating the landmarks within +/-4 pixels accuracy and we can therefore be confident that the system is quite stable to noise, well beyond the level normally present in this dataset. This is presumably because of the large degree of smoothing applied during the feature detection and correlation matching stages.

5 Discussion

This paper describes a system which can be trained from a few example images to automatically estimate the location of key features in 'veined' structures, such as insect wings. The performance of the automated system can be compared to human performance in terms of accuracy of landmark location. Although, the range of the system accuracy is nearly twice that of the manual digitization, the accuracy of the system can be considerably improved by using relevant models. The performance of the system is sufficiently accurate to allow it to replace the time consuming process of manual digitization, which is common to all morphometric studies.

The current system is capable of providing a set of 15 landmark locations on an image of size 1280 x 1022 pixels in about 3 minutes on a SUN Sparc Ultra 5 workstation. The time taken by a human to mark up one image using the standard mark-up tools is about 40 seconds though maintaining this speed across a large dataset might be regarded as unrealistic. We expect that the speed of the system can be optimized, however, the question of trade off between the speed and precision may arise (as more reference images may be needed to achieve the manual accuracy).

The pilot study can be scaled up with minor modifications and this automated method would be used in a scientific study with a large dataset comprising of 1600 images of different species of *Drosophila*. This analysis should enable us to test other performance aspects of the system, such as its reliability, and to evaluate any difficulties regarding the practical use of this dataset. The generic nature of object recognition and feature location incorporated in this automated system enables easy modification to locate features in a variety of other organisms. The method is intrinsically robust to changes in shape and based firmly on the statistical interpretation of data analysis. The system will be tested for its efficiency in locating the landmarks even in a scenario where the features to be located are quite complicated and beyond manual capabilities (eg. debris/bristles lying across one of the feature would be a major hurdle for manual digitization). Such an automated method will benefit major research groups in the morphometrics community and will easily be transferable to research groups in other relevant field of study. The automation of shape analysis has major potential advantages regarding standardization as the landmarks can be located without any manual intervention and will make large scale studies easily feasible [5, 8, 9]. The algorithms will be made available as an open source package via our website *www.tina-vision.net & www.flywings.org.uk*.

Acknowledgements

We would like to thank Dr Paul Bromiley & Dr Nicolas Navarro for their valuable comments on this paper.

References

[1] F.L. Bookstein. *Morphometric tools for landmark data: Geometry and Biology*. Cambridge University Press, UK, 1991.

[2] I.L. Dryden and K.V. Mardia. *Statistical Shape Analysis*. Chichester, John Wiley and Sons, 1998.

[3] J.P. Dujardin, C.B. Beard, and R. Ryckman. The relevance of wing geometry in entomological surveillance of triatominae, vectors of chagas disease. *Infection, Genetics and Evolution*, 7:161–167, 2007.

[4] J.M. Becerra and A.G. Valdecasas. Landmark superimposition for taxonomic identification. *Biological Journal of the Linnean Society*, 81:267–274, 2004.

[5] E. Zimmerman, A. Palsson, and G. Gibson. Quantitative trait loci affecting components of wing shape in drosophila melanogaster. *Genetics*, 155:671–683, 2000.

[6] C.P. Klingenberg and L.J. Leamy. Quantitative genetics of geometric shape in the mouse mandible. *Evolution*, 55:2342–2352, 2001.

[7] C.P. Klingenberg, L.J. Leamy, E.J. Routman, and J.M. Cheverud. Genetic architecture of mandible shape in mice: effects of quantitative trait loci analyzed by geometric morphometrics. *Genetics*, 157:785–802, 2001.

[8] A.S. Gilchrist, R.B.R. Azevedo, L. Partridge, and P. O'Higgins. Adaptation and constraint in the evolution of drosophila melanogaster wing shape. *Evolution & Development*, 2:114–124, 2000.

[9] C.P. Klingenberg, A.V. Badyaev, S.M. Sowry, and N.J. Beckwith. Inferring developmental modularity from morphological integration: analysis of individual variation and asymmetry in bumblebee wings. *American Naturalist*, 157:11–23, 2001.

[10] C.P. Klingenberg, L.J. Leamy, and J.M. Cheverud. Integration and modularity of quantitative trait locus effects on geometric shape in the mouse mandible. *Genetics*, 166:1909–1921, 2004.

[11] Klingenberg's Lab. www.flywings.org.uk.

[12] C.P. Klingenberg. Morphometrics and the role of phenotype in studies of the evolution of developmental mechanisms. *Gene*, 287:3–10, 2002.

[13] D. Houle, J.G. Mezey, P. Galpern, and A. Carter. Automated measurement of drosophila wings. *BMC Evolutionary Biology*, 3(25):1471–2148, 2003.

[14] D. Tharavy. Personal communication. *Paris National Museum of Natural History*, 2007.

[15] P.J.D. Weeks. Species-identification of wasps using principal component associative memories. *Image and Vision Computing*, 17:861–866, 1999.

[16] S. Palaniswamy, N.A. Thacker, and C.P. Klingenberg. A statistical approach to feature detection in digital images. *Leeds Annual Statistical Research Workshop*, pages 146–149, 2006.

[17] N.A. Thacker, P.A. Riocreux, and R.B. Yates. Assessing the completeness properties of pairwise geometric histograms. *Image and Vision Computing*, 13(5):423–429, 1995.

[18] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern analysis and Machine Intelligence*, 8(6):679–698, 1986.

[19] S. Palaniswamy, N.A. Thacker, and C.P. Klingenberg. A statistical framework for detection of connected features. *www.tina-vision.net/docs.php*, 2006.

[20] A.P. Ashbrook, N.A. Thacker, P.I. Rockett, and C.I. Brown. Robust recognition of scaled shapes using pairwise geometric histograms. *www.tina-vision.net/docs.php*, 1996.

Boosted Regression Active Shape Models

David Cristinacce and Tim Cootes
Dept. Imaging Science and Biomedical Engineering
University of Manchester, Manchester, M13 9PT, U.K.
david.cristinacce@manchester.ac.uk

Abstract

We present an efficient method of fitting a set of local feature models to an image within the popular Active Shape Model (ASM) framework [3]. We compare two different types of non-linear boosted feature models trained using GentleBoost [9]. The first type is a conventional feature detector classifier, which learns a discrimination function between the appearance of a feature and the local neighbourhood. The second local model type is a boosted regression predictor which learns the relationship between the local neighbourhood appearance and the displacement from the true feature location. At run-time the second regression model is much more efficient as only the current feature patch needs to be processed. We show that within the local iterative search of the ASM the local feature regression provides improved localisation on two publicly available human face test sets as well as increasing the search speed by a factor of eight.

1 Introduction

We describe a method of fitting a model of an object class to new images containing unseen examples. In this paper the class of objects is the human face, however the method can be applied to any type of object with corresponding features between different examples, for instance most types of medical images and many man-made objects.

This model based approach to computer vision requires a labelled set of training examples, with corresponding features between images (see Figure 1 for examples from our human face training set). There are many different types of models, most of which encode the appearance variation around or within the labelled region and also encode the shape variation of the feature locations across the training set [2, 3, 4, 5, 7].

This paper uses the Active Shape Model (ASM) framework due to Cootes *et al.* [3]. The ASM models shape variation across the training set with a statistical shape model and an individual model for each local feature. At run-time each local model updates its estimate of the best local match and the shape model is fitted to the full set of point estimates to eliminate false positive matches.

The original ASM paper [3] used local eigen patches [15] to model each feature. However in this paper we use non-linear boosted features trained using GentleBoost [9]. We investigate local feature detection using boosted features and also boosted regression, which aims to predict local feature points without the need for a sliding window search in the local neighbourhood. The boosted regression approach is shown to out perform local

feature detection when applied to the publicly available BIOID [10] and XM2VTS [13] data sets. The boosted regression approach is extremely fast, able to preform local search at > 60 frames per second and also able to achieve results comparable to other published methods [4].

2 Background

Active shape models are a method of modelling shape variation across a training set of labelled examples (see Cootes *et al.* [3]). The shape model can be fitted to a set of feature detections to remove outliers. There are various other shape constraint methods, such as the tree structure used in the Pictorial Structure Matching method due to Felzenszwalb and Huttenlocher [7] or the softer shape model constraint used by Cristinacce and Cootes [4] which take into account the local feature responses when fitting the shape model. However the ASM is a simple method which we use here to compare the performance of local regression versus detection models.

The choice of possible feature detection methods to use in the ASM is large. For example normalised correlation patches have been shown to be successful when combined with a generative model of appearance [4]. Other varieties of feature detectors are Local Binary Patterns [1], mutual information [5], Boosted Haar Wavelets [17] and K-Nearest Neighbour Classifiers [16]. The original ASM algorithm used local eigen models [15], but here we use discriminative haar wavelets trained using GentleBoost [9], as this technique has shown to work extremely well for whole face detection [12].

An alternative to feature detection methods for local search are regression techniques. For example the well known Active Appearance Model AAM algorithm [2] fits a deformable generative model to a patch of the image and then performs linear regression on the texture residual to update the internal model parameters and thus perform a local search. The AAM models the whole object, whereas our proposed method uses local features.

Another example of feature finding using a regression method is Zheng *et al.* [19], who use Rankboost [8] to rank the possible image warpings from the mean shape to the an unseen image and thus compute feature points. They present good results on manually cropped Echo Cardiograms and Face Photographs. Everingham *et al.* compare Kernel Ridge Regression with a Bayesian Classifier approach, but report better results with the simple classifier method for the task of eye finding [6].

A recent approach to using local regression models is described by Wimmer *et al.* [18], who train model trees to regress from local haar wavelet features to a objective function designed to peak at the true feature location. At run-time this allows the best matching location to be predicted for each feature. Langs *et al.* [11] use canonical correlation analysis to perform an AAM style search with filter responses located at individual feature points. Seise *et al.* [14] use the ASM framework in conjuction with a Relevance Vector Machine (RVM) regressor to update each feature location.

Our approach is similar to the approach of Wimmer and Seise, but uses GentleBoost as the regression function to predict the current displacement for each feature. We make a comparison between local regression methods and feature classifiers trained on the same data, both using the GentleBoost framework [9]. In Section 3 we describe our implementation in more detail and in Section 4 show that the regression method gives improved

localisation performance, compared to the boosted classifier, but at much lower computational cost.

3 Methodology

(a) Training Im 1 (b) Training Im 2 (c) Training Im 3 (d) Training Im 4

Figure 1: Manually Labelled Training Images

3.1 Active Shape Model

The Active Shape Model (ASM) was introduced by Cootes *et al.* [3] as a method of fitting a set of local feature detectors to an object and simultaneously taking into account global shape considerations. The allowable shape deformations are learnt from a manually labelled training set (see Figure 1) to produce a linear shape model with the following form:-

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}_s \mathbf{b}_s \tag{1}$$

Where $\bar{\mathbf{x}}$ is the mean shape, $\mathbf{P}_s$ is a set of orthogonal modes of variation and $\mathbf{b}_s$ is a set of shape parameters. Given a set of hypothesised feature points $\mathbf{Y}$ in the image plane the shape model parameters $\mathbf{b}_s$ can be determined by minimising

$$|\mathbf{Y} - T_{\mathbf{t}}(\bar{\mathbf{x}} + \mathbf{P}_s \mathbf{b}_s)| \tag{2}$$

By placing constraints on the the allowable shape parameters $\mathbf{b}_s$ the shape model estimate of the current feature points $T_{\mathbf{t}}(\bar{\mathbf{x}} + \mathbf{P}_s \mathbf{b}_s)$ are constrained to form a plausible shape.

The shape model is active in the sense that feature detectors are applied to search in the local neighbourhood of each point and the best match of each detector is recorded. Assuming the majority of the detections are correct, the shape model can be fitted to this set of points and outlier detections discarded. This constraint on feature matching has been shown to improve results compared to merely taking the best unconstrained fit of each feature [3].

3.2 Boosted Feature Detection

Any set of feature detectors can be used in the ASM framework described above. The original algorithm [3] used eigen model [15] profiles of the texture about each of the individual feature points. In this work we choose boosted feature detectors, which have

a similar formulation to the well known Viola and Jones face detector [17]. The training method we use is GentleBoost [9], which has shown to give superior performance compared to the original AdaBoost algorithm for the task of face detection [12].

Algorithm 1 Gentle Boost Training Algorithm - Classification [9]

1. Start with weights $w_i = 1/N$, $i = 1,2,...,N$, $F(x) = 0$ and $y_i = 1$ for positive examples, $y_i = -1$ for negative examples.
2. Repeat for $m = 1,2,...M$:
   (a) Fit all the regression functions $f_m(x)$ by weighted least squares of y_i to x_i with weights w_i.
   (b) Select the $f_m(x)$ with least weighted error $\sum_{i=1}^{N}(w_i(y_i - f_m(x_i)))^2$
   (c) Update $F(x) \leftarrow F(x) + f_m(x)$
   (d) Update $w_i \leftarrow w_i exp(-y_i f_m(x_i))$ and re-normalise
3. Output the classifier $sign[F(x)] = sign\left[\sum_{m=1}^{M} f_m(x)\right]$

The GentleBoost classifier training procedure is described in Algorithm 1. The aim of the algorithm is to learn a discrimination function between a set of positive and negative examples. Where positive examples are image patches centred on the correct feature locations and negative examples are nearby examples displaced from the true locations, see Figure 2.

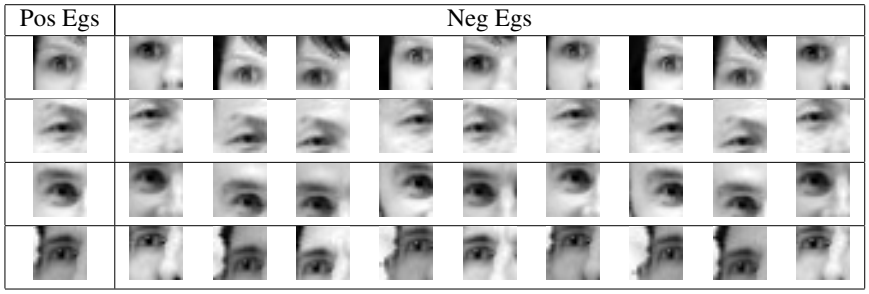


Figure 2: Positive and Negative Examples for right eye detector

Following the notation in Algorithm 1, each positive training patch x_i has label $y_i = +1$ and each negative patch has label $y_i = -1$. To train using GentleBoost it is necessary to select a family of functions $f(x)$ which take an image patch x_i and attempt to predict the classification y_i for a given set of training weights w_i. In this paper $f(x)$ is a binned histogram of responses from a haar wavelet (we use the same set as [17]). Each $f(x)$ is trained by computing the weighted mean of target values y_i in each histogram bin. The error for each $f(x)$ is the weighted sum of square differences between the target value y_i and the mean of the selected bin determined by the wavelet response to patch x_i. The GentleBoost training algorithm selects a set of weak classifier functions and outputs a

strong classifier, as described in Algorithm 1. The training algorithm is computationally expensive, as the weak classifier functions $f(x)$ have to be retrained at every iteration with new weights w_i.

There are also several parameters that need to be set before training can take place, namely the resolution of the image patch, which determines the number of potential haar wavelet weak classifiers $f(x)$, the number of training rounds M, the number of histogram bins h_b and the number of training example patches N. With 21x21 pixel image patches, $N = 179,545$ (1205 positive patches, 178,340 negative patches), $M = 200$, $h_b = 25$ the training for each patch completes in $\sim$ 20hrs on a single node of a 64bit multi-processor cluster running Linux. The feature models are trained independently therefore the whole model can be built in $\sim$ 20hrs, if enough nodes are available. The parameters above are unlikely to be optimal. For example, it may well be possible to improve the results, by increasing the number of training rounds M or increasing the size of the training set, which currently only consists of 1205 face images (see Figure 1).

3.3 Boosted Feature Regression

In the feature detection approach described in Section 3.2 the model is trained on positive examples centred on a small neighbourhood around manually labelled feature locations. Negative examples are feature patches displaced from the true locations (see Figure 2).

However an obvious problem with feature detector training is where to draw the boundary between positive examples and nearby false examples. In Section 3.2 we take a conservative approach and only treat image patches centre on the true feature as positive examples. Patches between 1 pixel and 3 pixels away are treated as ambiguous, while patches greater than 5 pixels away are classed as false positives (a similar approach is adopted in [6]).

This works reasonably well, but is arbitrary and also throws away potentially useful information, such as the distance of each patch from the true positive. An alternative technique which makes use of this information is regression, which learns the relationship between the displacement to the true feature location and the textural appearance of the local neighbourhood around each feature point.

Algorithm 2 Gentle Boost Training Algorithm - Regression [9]

1. Start with input values x_i and target values y_i for $i = 1,2,...,N$ and $F(x) = 0$ and small positive constant α.
2. Repeat for $m = 1,2,...M$:
   (a) Fit the regression function $f_m(x)$ by least squares of y_i to x_i.
   (b) Select the $f_m(x)$ with least error $\sum_{i=1}^{N}(y_i - f_m(x_i))^2$
   (c) Update $F(x) \leftarrow \alpha F(x) + f_m(x)$
   (d) Update the residual target value $y_i \leftarrow y_i - f_m(x_i)$
3. Output the regression function $F(x) = \sum_{m=1}^{M} f_m(x)$

We use the GentleBoost logistic regression method described by Friedman *et al.* [9]

(see Algorithm 2) which has very similar form to the GentleBoost classification training method use in Section 3.2. The same image patches x_i are used as in the classification training, however instead of y_i being class labels $\{-1,+1\}$ they are local displacement values in the training image frame (suitably scaled - see Figure 3). The regression training therefore uses the whole training data available, whilst the classification training discards some ambiguous patches (marked with an X in Figure 3).

Image									
Regression	-4	-3	-2	-1	0	+1	+2	+3	+4
Class	-1	-1	X	X	+1	X	X	-1	-1

Figure 3: Examples of training patches for right eye from one of the training images (depicting translation in the x-coordinate). Regression training values are the displacement from the centre of the patch to the true eye pupil location (shown by a white cross). Classification training values are -1 for negative examples, +1 for positive examples, images marked with a X are ignored during classifier training

The GentleBoost regression algorithm then proceeds as described in Algorithm 2. The Haar wavelet functions $f(x)$ are fitted to the weighted training patches x_i with displacement y_i. A function $f(x)$ is selected at each stage, the residual displacements y_i are adjusted and after M rounds a strong regressor function $F(x)$ is output.

Note that in Algorithm 1 the weights on each training example w_i are updated between training rounds. In Algorithm 2 the target values y_i vary between boosting rounds and the training examples have equal weight. Another important difference between the classification algorithm and the regression method is that the regression requires two models per feature point to predict the x and y displacements for each patch. The training time for each regression model is also increased slightly due to the extra training samples close to the true feature points being included in the training set (which are discarded during classifier training).

An additional parameter of the regression training algorithm is α which represents the learning rate. This can be any value in the approximate range $[0.1, 1.0]$ and needs to be chosen apriori. The value can be shown to be equivalent to the shrinkage parameter in Lasso Regression [9]. Small values of α result in slower training, but more diverse feature selection. We set $\alpha = 0.25$ in our experiments.

3.4 Summary of Method

At run-time the search proceeds as follows:-

1. Find initial feature points - for example using a global detection method
2. Iterate the following:-
   (a) Search around the current feature location with a feature detector - Or alternatively predict the improved feature location using boosted regression
   (b) Fit the shape model to the current set of feature locations to remove outliers

   Until Converged.

4 Experiments

4.1 Test Criteria

The models described in Section 3 are applied to two publicly available test sets, with manually labelled ground truth, namely the BIOID [10] and XM2VTS [13] data sets. The criteria for success is the distance of the points computed using automated methods compared to manually labelled ground truth. The distance metric is shown in Equation 3.

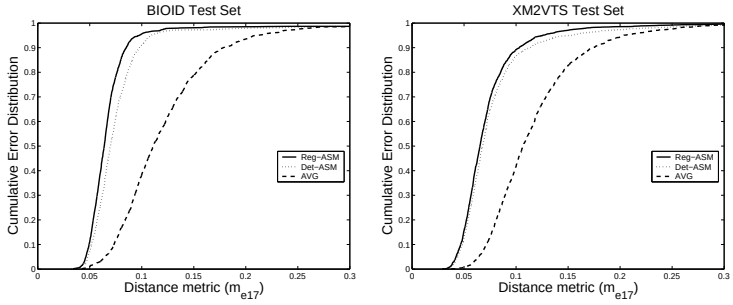
$$m_e = \frac{1}{ns}\sum_{i=1}^{i=n} d_i \quad (3)$$

Here d_i are the Euclidean point to point errors for each individual feature location and s is the ground truth inter-ocular distance between the left and right eye pupils. $n = 17$ as only the internal feature locations around the eyes, nose and mouth are used to compute the distance measure. The five feature points on the edge of the face (see Figure 1) are ignored for evaluation purposes, due to their high variability between different human annotators.

4.2 Full Search Results

The fully automatic search is investigated on the two faces test sets. Three separate procedures are investigated as follows:-

- AVG - Average points within the global Viola and Jones face detector (dashed line)
- Det-ASM - Detection Features and Active Shape Model, initialised with the average points (dotted line)
- Reg-ASM - Regression Features and Active Shape Model, initialised with the average points (solid line)



(a) BIOID (b) XM2VTS

Figure 4: Cumulative distribution of point to point error measure on XM2VTS and BIOID test sets when using face detection to initialise the local search

Figure 4 shows that the Reg-ASM and Det-ASM give similar results on both the BIOID and XM2VTS data sets. Both local search methods combined with the ASM give

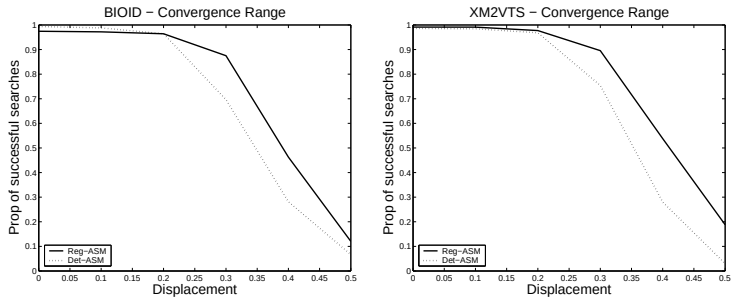
a large improvement relative to the average points found within the global face detector window. For example on the BIOID data set (see Figure 4(a) - dashed line) 75% of faces have a point to point error $m_e < 0.15$ using the average points. However after the local ASM search is applied 95% of faces are found at this accuracy limit (see solid line). The Reg-ASM method performs slightly better than the Det-ASM on both data sets (compared solid+dotted lines in Figure 4).

The results in Figure 4 are comparable with the authors previous results published on the two data sets. For example using the same error measure on the BIOID data set the Constrained Local Model (CLM) algorithm [4] gives a 90% success rate at $m_e < 0.1$ compared to 95% using the prorposed Reg-ASM algorthim. For lower values of m_e the CLM is more accurate, however the Reg-ASM and Det-ASM methods described here are initialised using the average points from the face detector. In [4] the Pictorial Structure Matching(PSM) algorithm [7] is used as part of a three stage method. The Reg-ASM local search is also much more efficient than the CLM algorithm (see Section 4.4).

4.3 Displacement Results

In order to determine the range of convergence of the Reg-ASM and Det-ASM the tracking methods are systematically displaced from the true feature locations in the eight possible compass directions, by a percentage of the inter-ocular distance and the shape reset to the mean of the statistical shape model.

This gives a total of 8 starting search locations per image, to start the Reg-ASM and Det-ASM algorithms, at each of five possible displacements of 10%, 20%, 30%, 40% and 50% of the inter-ocular distance. The rate of convergence for the Reg-ASM and Det-ASM given a point to point error limit of $m_e < 0.15$, for this range of displacements is shown in Figure 5.



(a) BIOID (b) XM2VTS

Figure 5: Range of convergence for regression an detection methods on XM2VTS and BIOID test sets

Figure 5 shows that the Reg-ASM has a wider range of convergence compared to the Det-ASM on both the BIOID and XM2VTS data sets. This is possibly due to the regression prediction for each point (in the Reg-ASM) being able to jump over false minima which may be found by the Det-ASM search. However both algorithms have some in-built ability to avoid false minima due to the ASM shape fitting step which removes outlying predictions for individual feature points.

4.4 Timings

The local search time using the Reg-ASM and the Det-ASM methods is dependent on the search image and the starting displacement. However both algorithms converge in fewer than 5 iterations in most cases. Therefore the search speed is dependent on the time for one iteration of the ASM.

In our implementation one iteration of the Reg-ASM takes $\sim$3ms compared to $\sim$ 25ms with the Det-ASM[1], using a C++ implementation on a P4 3GHz processor. Therefore the Reg-ASM is approximately eight times quicker then the Det-ASM. If 1-5 iterations are required when tracking a face with the Reg-ASM in a video sequence the frame rate will be approximately 60-300 frames per second.

(a) Start Pts (b) After It 4 (c) After It 8 (d) Final Pts

Figure 6: Example of Iterative Reg-ASM Search on BIOID image

5 Conclusions

We have compared two local feature updates methods within the Active Shape Model framework. The boosted regression approach is shown to have a wider range of convergence compared to the boosted classifier method on two publicly available face data sets. The boosted regression method is also more computationally efficient by a factor of eight, which makes it suitable for use in real time systems.

Future work will involve building larger models with more data and different data sets. We are particularly interested in applying the boosted regression approach to high dimensional medical images, as in more than two dimensions the feature detection search at run-time becomes prohibitively expensive. We may also apply the regression update step in other formulations such as the AAM.

The boosted regression feature prediction method described is an extremely efficient local search algorithm ($>$ 60 frames per second), which improves on standard boosted feature detection approaches. We anticipate that this form of boosted regression update will be useful in other areas of computer vision.

[1]Note it may be possible to improve the efficiency of the Det-ASM by introducing a cascade structure for each classifier as in [17]. However the fact that the classifier has to search the local neighbourhood will always make it slower than the regression model, if both methods use the same number of weak learners.

References

[1] T. Ahonen, A. Hadid, and M. Pietikainen. Face recognition with local binary patterns. In 8^{th} *European Conference on Computer Vision 2004, Prague, Czech Republic*, pages 469–481, 2004.

[2] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In 5^{th} *European Conference on Computer Vision 1998, Freiburg, Germany*, volume 2, pages 484–498, 1998.

[3] T. F. Cootes and C. J.Taylor. Active shape models. In 3^{rd} *British Machine Vision Conference 1992*, pages 266–275, 1992.

[4] D. Cristinacce and T. Cootes. Detection and tracking with constrained local models. In 17^{th} *British Machine Vision Conference 2006, Edinburgh, Scotland*, pages 929–938, 2006.

[5] N. Dowson and R. Bowden. Simultaneus modeling and tracking (smat) of feature sets. In 23^{rd} *Computer Vision and Pattern Recognition Conference 2005, San Diego, USA*, pages 99–105, 2005.

[6] M. Everingham and A. Zisserman. Regression and classification approaches to eye localistion in face images. In 7^{th} *International Conference on Automatic Face and Gesture Recognition 2006, Southampton, UK*, pages 441–446, 2006.

[7] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61:55–79, 2005.

[8] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, 2003.

[9] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *The Annals of Statisics*, 28:337–407, 2000.

[10] O. Jesorsky, K. J. Kirchberg, and R. W. Frischholz. Robust face detection using the hausdorff distance. In 3^{rd} *International Conference on Audio- and Video-Based Biometric Person Authentication 2001, Halmstad, Sweden*, pages 90–95, 2001.

[11] G. Langs, P.Peloschek, R. Donnner, M. Reiter, and H. Bischof. Active feature models. In 18^{th} *International Conference on Pattern Recognition 2006, Hong Kong, China*, pages 417–420, 2006.

[12] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *IEEE International Conference on Image Processing*, pages 900–903, New York, USA, 2002.

[13] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre. Xm2vtsdb: The extended m2vts database. In 2^{nd} *International Conference on Audio- and Video-Based Biometric Person Authentication 1999, Washington DC, USA*, pages 72–77, 1999.

[14] M. Seise, S. McKenna, I. W. Ricketts, and C. A. Wigderowitz. Learning active shape models for bifurcating contours. *IEEE Transactions on Medical Imaging*, 26(5):666–677, 2007.

[15] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[16] B. van Ginneken, A.F. Frangi, J.J. Staal, B.M. ter Haar Romeny, and M.A. Viergever. Active shape model segmentation with optimal features. *IEEE Transactions on Medical Imaging*, 21:24–933, 2002.

[17] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In 19^{th} *Computer Vision and Pattern Recognition Conference 2001, Hawaii, USA*, volume 1, pages 511–518, Kauai, Hawaii, 2001.

[18] M. Wimmer, F. Stulp, S. Tschechne, and B. Radig. Learning robust objective functions for model fitting in image understanding applications. In 17^{th} *British Machine Vision Conference 2006, Edinburgh, Scotland*, pages 1159–1168, 2006.

[19] Y. Zheng, X. S. Zhou, B. Georgescu, S.K. Zhou, and D. Comaniciu. Example based non-rigid shape detection. In 9^{th} *European Conference on Computer Vision 2006, Grax, Austria*, pages 423–436, 2006.

Tracking Through Clutter Using Graph Cuts

James Malcolm Yogesh Rathi Allen Tannenbaum
School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, Georgia 30332-0250
{malcolm,yogesh.rathi,tannenba}@bme.gatech.edu

Abstract

The standard graph cut technique is a robust method for globally optimal image segmentation. However, because of its global nature, it is prone to capture outlying areas similar to the object of interest. This paper proposes a novel method to constrain the standard graph cut technique for tracking objects in a region of interest. By introducing an additional penalty on pixels based upon their distance from a region of interest, segmentation is biased to remain in this area. We employ a filter which predicts the location of the object. The distance penalty is then centered at this location and adaptively scaled based on prediction confidence. This method tracks at real-time rates and easily generalizes to tracking multiple noninteracting objects.

1 Introduction

Tracking rigid objects has been the focus of much research, and the problems accompanying this key task are well-known. For example, the object might have weak edges causing the segmentation to leak out into the surrounding area, or the object may move suddenly outside the algorithm's region of detection, or the object may be near other objects of similar intensity causing unintended objects to be tracked.

Various methods have been proposed to overcome these difficulties. To keep segmentations from spilling over object boundaries, learned shape priors constrain segmentation to a set of possible shapes [8, 9, 14]. To account for object movement, motion models can predict the likely location of the object in subsequent frames [7, 11]. When adjacent regions are similar to the object of interest, multiple hypothesis trackers can keep track of each region while determining the most likely in each frame based on some criteria [1, 12, 15, 18].

1.1 Graph cut techniques

Graph cut techniques have received considerable attention as robust methods for image segmentation. Despite their widespread use for computer vision problems such as image segmentation and stereo disparity, graph cuts have received little attention with respect to tracking. This is largely due to the global segmentations they produce which tend to catch unintended regions that are similar to the object of interest. For example, the standard graph cut technique for image segmentation [4] finds regions with high likelihood given intensity priors. Figure 1 shows an example where there are multiple regions of similar

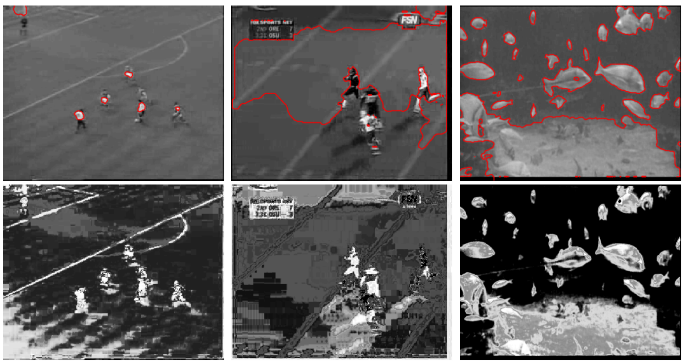


Figure 1: Standard graph cut segmentation *(top)* and normalized likelihood of object intensity used in graph edge weights *(bottom)*. Likely regions throughout the image are captured with the standard method making it unsuitable for tracking.

intensity to the object. The standard graph cut algorithm captures such regions. Post-processing must be performed to filter out those regions that are not part of the object. However, this same feature, that of grabbing such regions anywhere in the image, naturally solves the problem of large object movements. The graph cut will find the object even if it moved far relative to its location in the previous frame. The problem is now one of constraining the graph cut to capture only the object of interest, even if it made a large movement yet ignoring other regions of similar intensity. Hence a spatial constraint is needed.

Several techniques have used graph cuts for segmentation in visual tracking applications. In [22] the segmentation is constrained to a narrow band. For each frame, successive graph cut segmentations converge on a final segmentation, each pass constrained to a narrow band around the cut boundary resulting from the previous pass. This method is dependent upon initial contour placement and requires repeated cuts on this reduced domain. In [10] the authors use one graph cut for each frame to both estimate the optical flow and object position based on that flow despite changes in illumination. However, since optical flow requires the multi-label graph cut technique [6] and the graph proposed has such dense neighborhoods, the authors' current approach requires about a minute per frame. Also, due to the local nature of optical flow, the technique cannot handle large movements.

Besides tracking, work has been done to constrain segmentations based on a user selected region. The work of [19] begins with a rectangle bounding the object, while the work of [2] uses a narrow band to constrain segmentation. Both perform successive graph cut segmentations incorporating additional user interaction with each pass. Neither method is targeted towards tracking *per se*, but instead seeks a perfect segmentation. In these works, hard constraints confine the segmentation within a user-selected region and multiple graph cuts are performed. In our work, the object may be found a distance from the predicted centroid depending on the scale of the distance penalty, and segmentation is performed only once per frame.

1.2 Our contributions

The method presented here makes several important contributions to the field of visual tracking. First, we incorporate a distance penalty into the graph cut algorithm to bias segmentations to a region likely to contain the object. Second, we present a simple filter to predict the object location based on the centroid of the previous segmentation and a moving average of the object's velocity. The distance penalty is then centered at the predicted object centroid and extends outward forming a basin of attraction. Third, to further integrate the filter with the distance penalty, the scale of this distance penalty, and hence the slope of its surface, is adaptively set based on the prediction error. Finally, since the segmentation is performed in one cut using the standard binary label graph cut method, the unoptimized system tracks at up to 15 Hz on 240x320 images using a Pentium IV 3.6 GHz workstation. The method generalizes to multiple noninteracting objects.

The rest of the paper is organized as follows. Section 2 outlines the standard graph cut segmentation framework. Section 3 describes the distance penalty constraining segmentation. Section 4 defines the filter used to predict the object centroid. Section 5 integrates the filter prediction error with the distance penalty. Next, in Sections 6 and 7, we present our algorithm and results on several video sequences tracking single and multiple objects. Finally, in Section 8 we summarize our work and describe some possible future research directions.

2 Graph cuts

In this section, we briefly outline the graph cut methodology; for more details see [2, 3, 4, 19] and the references therein. Taking advantage of efficient algorithms for global min-cut solutions, we cast the energy-based image segmentation problem in a graph structure of which the min-cut corresponds to a globally optimal segmentation.

Evaluated for a pixel object/background assignment A, such energies are designed as a data dependent term and a smoothness term. The data dependent term evaluates the penalty for assigning a particular pixel to a given region. The smoothness term evaluates the penalty for assigning two neighboring pixels to different regions, i.e. a boundary discontinuity. These two terms may be thought of as a region-based term and a boundary term, often weighted by $\lambda \geq 0$ for relative influence:

$$E(A) = \sum_{p \in I} R_p(A_p) + \lambda \sum_{\substack{(p,q) \in \mathcal{N} \\ A_p \neq A_q}} B_{(p,q)} \tag{1}$$

where I represents all image pixels, $\mathcal{N}$ all unordered neighborhood pixel pairs. The choice of neighborhood size and structure has a large influence on the solution as smaller neighborhoods tend to introduce metrication artifacts [5].

To construct the graph representing this energy, each pixel is considered as a graph node in addition to two nodes representing object and background. The data dependent term is implemented by connecting each pixel to both the object and background nodes with non-negative edge weights $R_p(\mathcal{O})$ and $R_p(\mathcal{B})$ representing the penalty for assigning pixel p to the object or background region, respectively. Lastly, the smoothness term is implemented by connecting each pairwise combination of neighboring pixels (p,q) with a non-negative edge weight $B_{(p,q)}$ representing the penalty for separating pixels p and q.

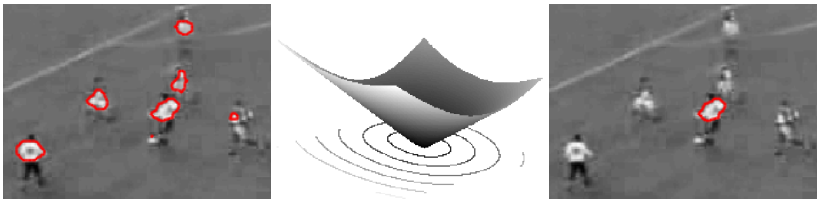


Figure 2: Mean intensity tracking of a soccer player among others of similar intensity: no distance penalty, distance penalty ϕ with isocontours, applying distance penalty *(left to right)*. Without the distance penalty, multiple non-intended objects were captured.

Notice that, since the min-cut sums only along the boundary, the boundary condition of $A_p \neq A_q$ in (1) may be ignored and every pair of neighboring pixels may be connected with edge weight $B_{(p,q)}$. The min-cut of the weighted graph represents the segmentation that best separates the object from its background. See [4] for more details.

Typical applications of graph cuts to image segmentation differ only in the definitions of R_p and $B_{(p,q)}$. For example, the authors of [4] use the negative log-likelihood of a pixel's intensity to compute the regional weights while intensity contrast is used in the boundary term:

$$R_p(O) = -\ln P(I_p|O), \quad R_p(\mathcal{B}) = -\ln P(I_p|\mathcal{B}), \quad B_{(p,q)} = \exp\left(\frac{-\|I_p - I_q\|^2}{2\sigma^2}\right)\frac{1}{\|p-q\|} \tag{2}$$

where $\|p-q\|$ is the standard L_2 Euclidean norm yielding pixel distance in the image and σ^2 is often set to the average squared norm: $\sigma^2 = \frac{1}{|\mathcal{N}|}\sum_{(p,q)\in\mathcal{N}}\|I_p - I_q\|^2$. In [4] the user marks regions of object and background that are then used to generate the intensity histograms for calculating $P(I_p|O)$ and $P(I_p|\mathcal{B})$ (see Figure 6).

The authors of [24] demonstrate the use of the mean intensity of the two regions to classify image pixels into two piecewise constant regions. They propose the following definitions:

$$R_p(O) = (I_p - \mu_O)^2, \qquad R_p(\mathcal{B}) = (I_p - \mu_{\mathcal{B}})^2, \qquad B_{(p,q)} = \frac{c_{\mathcal{N}}}{\|p-q\|} \tag{3}$$

where μ_O and $\mu_{\mathcal{B}}$ are the mean intensities of the regions marked by the user as object and background and $c_{\mathcal{N}}$ is a constant based on the chosen neighborhood size.

3 Distance penalty

The standard graph cut technique is capable of finding regions matching the object intensity located anywhere in the image. By penalizing pixels based on their distance from the expected location, a potential well is formed biasing segmentation to a region of interest. Figure 2 shows segmentation with and without such a penalty in the presence of multiple similar objects.

The distance penalty ϕ is formed from the user segmented shape of the object in the first frame. Centering that mask M at the predicted object location and assigning it zero penalty, each pixel x outside the mask is assigned its distance from the nearest masked pixel $m_x \in M$, i.e. $\phi(x) = \|x - m_x\|$ or zero if $x \in M$. Such a construction can be quickly computed with the Fast Marching algorithm [20, 23]. More deformable shape priors may be used for the base patch [10, 17, 21].

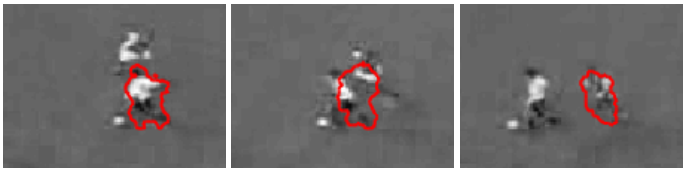


Figure 3: Without location prediction, tracking can fail when the target makes sudden movements. Here the tracker catches a defender as the target passes *(left to right)*.

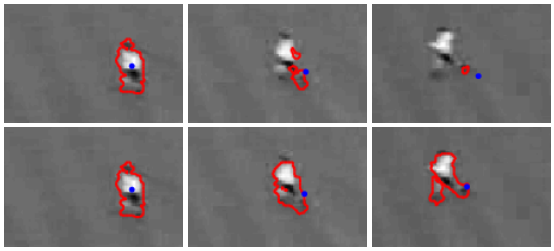


Figure 4: Effect of adaptive α on full intensity tracking: non adaptive alpha (assume zero error) *(top, left to right)*, alpha with prediction error *(bottom, left to right)*. Tracking fails without using error feedback to scale distance penalty.

4 Location prediction

It is often the case that the object makes a large movement, large enough at times to place it in an area of high distance penalty. To overcome this problem, we predict the location of the object in each frame based on its previous location and center the distance penalty at this predicted location.

To demonstrate the need for some form of prediction, we experimented with the assumption that the object has not moved: the distance penalty is centered at the last known object position. Figure 3 shows the failure to track after the object has made a sudden move, despite the use of adaptive α scaling described in Section 5. The movement placed the object too far outside of the basin of attraction.

Introducing actual prediction, we assume the object is traveling with continuous velocity, hence we predict the next object location $\tilde{c}_{t+1}$ based on projecting forward by the average displacement in the past few frames. A simple filter that projects the centroid c_t forward in time based on a moving average of the past N displacements is defined as:

$$\tilde{c}_{t+1} = c_t + \frac{1}{N}\sum_{j=0}^{N}(c_{t-j} - c_{t-j-1}). \tag{4}$$

5 Error feedback

We now have the distance penalty constraining segmentation and the filter predicting where to center this distance penalty, but what if the filter is wrong? Figure 4 shows such a case. The object has made a sudden move outside the predicted basin of attraction.

What is needed is a way of adaptively scaling the distance penalty based on the prediction error. In this work, we take the error in prediction to be the distance between the

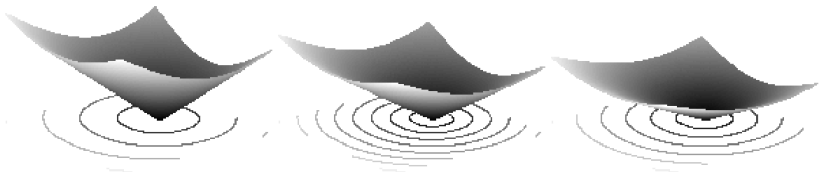


Figure 5: Distance penalty surface and isocontours are shown scaled by α for increasing prediction error $\|\tilde{c}-c\|$. Notice the basin of attraction widening as the error increases *(left to right)*.

Figure 6: User initialization of object *(red)* and background *(blue)* regions: original and initialization scribbles *(left to right)*.

predicted $\tilde{c}$ and actual c centroids. The distance map is then scaled by $\alpha(\|\tilde{c}-c\|)$ taken from an exponential distribution of the prediction error, $\alpha(x) = \exp\left(-x^2/\rho^2\right)$, where ρ is user specified based on empirical motion. The effect is that when the filter is off in its predictions of the object centroid, the distance penalty is lowered to hopefully still capture the object. After locking back onto the object, the α automatically raises the distance penalty back up to tighten around the object as the error decreases. See Figure 5 for a visual of this distance penalty as it is scaled by α for increasing prediction error. Figure 4 shows how, despite incorrectly predicted centroids, the system is able to recover by adaptively widening the distance penalty.

6 Proposed algorithm

In an observer-type framework, at each frame the algorithm predicts the object location, determines the distance penalty scaling based on prediction error, computes edge weights for the graph, and performs a graph cut segmentation. For initialization, the user is required to roughly mark in the first frame the object and background as in Figure 6. This initialization defines the intensity priors used in constructing the priors used in regional edge weights (2) and (3).

In the prediction step, the centroid from the previous frame's segmentation is used as a measurement c. The filter predicts the object centroid location in this new frame $\tilde{c}$ from a moving average of displacements as in (4).

The $\alpha(\cdot)$ scaling function for the distance penalty is calculated from an exponential distribution of error $\|\tilde{c}-c\|$. Since the proposed simple filter is unstable against large displacements, we found the need to limit this distance in practice to a user-defined γ so that the distance penalty is not driven completely to zero. The $\alpha(\cdot)$ used is then:

$$\alpha(x) = \exp\left(\frac{-\min(x,\gamma)^2}{\rho^2}\right). \tag{5}$$

We propose a new regional edge weight to augment the standard weights in (2) and

(3). Our goal is to determine $P(O|I)$ for each pixel, and Bayes rule tells us that $P(O|I) \propto P(I|O)P(O)$. If we were to assume $P(O)$ and $P(\mathcal{B})$ are uniform, then their negative log-likelihoods are zero, and so they fall out of the expression as in (2). Here, we assume a non-uniform object prior $P(O)$ and claim: $-\ln P(O) \propto \alpha(\|\tilde{c}-c\|)\phi$. We assume the background to still be uniformly distributed $P(\mathcal{B})$. Introducing a weight $\beta \geq 0$ for relative distance penalty influence, we have a new regional term:

$$R_p(O) = -\ln P(I_p|O) - \beta \ln P(O_p) = -\ln P(I_p|O) + \beta\, \alpha(\|\tilde{c}-c\|)\phi(p) \quad (6)$$

$$R_p(\mathcal{B}) = -\ln P(I_p|\mathcal{B}) - \beta \ln P(\mathcal{B}_p) = -\ln P(I_p|\mathcal{B}) \quad (7)$$

Similarly, this additional weight may be added to the regional mean intensity term (3):

$$R_p(O) = (I_p - \mu_O)^2 + \beta\, \alpha(\|\tilde{c}-c\|)\phi(p) \quad (8)$$

$$R_p(\mathcal{B}) = (I_p - \mu_{\mathcal{B}})^2. \quad (9)$$

We use the standard intensity contrast smoothness term (2) for all experiments. Finally, we take the min-cut of this graph to yield a binary segmentation.

To track multiple similar objects, the same distance penalty may be used if the objects do not interact. If the objects do not touch, then their respective potential wells separate the segmentation into blobs. The centroid of each object is predicted independently. Since the segmentation is a binary mask of indistinguishable blobs, the identity of each blob is assigned to the object of nearest centroid in the previous frame. See Figure 11 for an example of simultaneously tracking two soccer players. If the objects were to touch, the segmentation will likely merge blobs and the unique identity of such blobs would be undefined for determining object centroids.

7 Results

Tracking was performed on three natural image sets and representative frames chosen to exhibit clutter with objects of similar intensity. Full videos are included in the supplementary material. The system is a combination of Matlab and C/C++ operating on a Pentium IV 3.6 GHz processor with 2GB RAM and tracks at roughly 5-15 Hz depending upon the graph neighborhood used[1]. The image size in the fish sequence is 360x480 while both the soccer and football sequences have frames of size 240x320.

Parameters are defined as follows. For all experiments, objects are assumed to not move more than 5 pixels between frames so $\gamma = 5$ in (5) and in practice $\rho = \frac{1}{2}\gamma$ is quite robust. For all full intensity experiments, $\lambda = 6$ in (1) and $\beta = 8$ in (6). For all mean intensity experiments both $\lambda = 10000$ and $\beta = 10000$.

The choice of neighborhood directly influenced the speed of computing the graph cut since larger neighborhoods induced denser graphs. Using a neighborhood of size 4 enabled tracking at 15 Hz, size 8 at 9 Hz, and size 16 at 5 Hz. The choice of neighborhood also affects the smoothness of the segmentation. Smaller neighborhoods tend to introduce irregular segmentations [5]. It is important to note that, since the segmentations for sizes 4 and 8 were not as smooth, they introduced larger variations in the calculated centroid and hence larger prediction errors. Increased smoothing (λ) was required to maintain track

[1]The min-cut is computed using the publicly available software of Vladimir Kolmogorov (`http://www.adastral.ucl.ac.uk/ vladkolm/`)

Figure 7: Several frames from the soccer sequence using full intensity capturing more of the multimodal object. Target object makes contact with another player yet the filter breaks them free. Full image *(left)* and selected cropped frames *(right)*. Yellow dot represents predicted centroid.

Figure 8: Several cropped frames from the soccer sequence using mean intensity. Blue dot represents predicted centroid.

with smaller neighborhoods. Tracking with size 4 or 8 was therefore not as robust as size 16. Unless otherwise noted, results are shown with a neighborhood of size 16.

The first video sequence involves several soccer players of similar intensity. Figure 7 shows full intensity tracking grabbing much of the object while Figure 8 shows mean intensity tracking grabbing the bright jersey, the optimal piecewise constant segmentation.

The second video sequence involves two dark football players touching. Figure 9 shows that despite this, the filter is able to track the intended player.

The third video sequence involves a fish crossing the screen among many other fish of identical intensity distributions. The high frame rate of the video sequence results in the fish moving slowly resulting in extended contact with the other fish. Figure 10 shows several such frames where the distance penalty correctly contains the segmentation.

In Figure 11 we demonstrate mean intensity tracking of multiple similar noninteracting objects.

8 Conclusion

This paper demonstrates a distance penalty to constrain the standard graph cut segmentation to a region of interest. An observer is proposed to predict object location while the prediction error is used to scale the distance penalty forming a basin of attraction that is adaptively sized. The binary graph cut algorithm is then used to find the object in one pass. The method operates at real-time rates and generalizes to multiple noninteracting targets.

There are several future directions of research. The multi-label graph cut method [6] may naturally allow segmentation of multiple dissimilar objects with interaction penalties. Anisotropic distance penalties may be used to bias certain directions based on expected

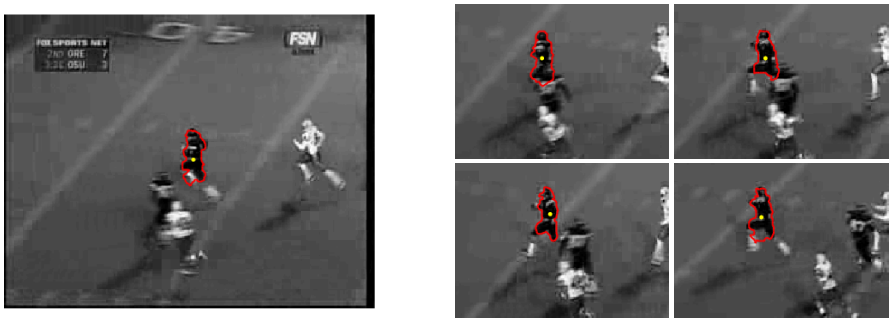


Figure 9: Several frames from the football sequence showing the target touching a teammate yet maintaining track *(yellow dot)*. Full image *(left)* and selected cropped frames *(right)*.

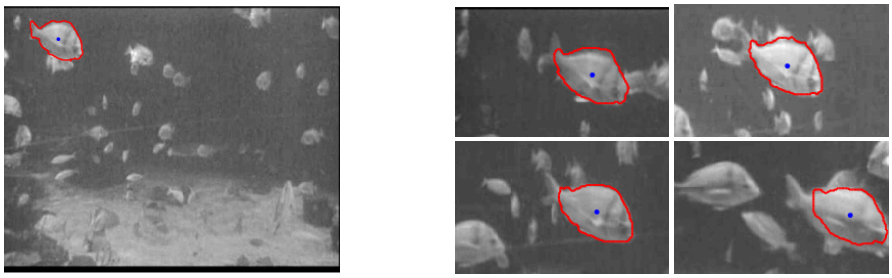


Figure 10: Selected frames from the fish sequence where the segmentation is correctly contained despite prolonged contact with other fish of similar intensity. The fish accelerates toward the end of the sequence yet the filter manages to maintain track *(blue dot)*. Full image *(left)* and selected cropped frames *(right)*.

object trajectory. Instead of rebuilding the graph from scratch for each frame as in the current system, speed can be enhanced by updating the graph in place from frame to frame [13]. Furthermore, segmentation may be made more robust for a larger class of imagery by tracking in a feature space with more information than simple intensity [16].

References

[1] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear non-gaussian bayesian tracking. *IEEE Trans. Signal Processing*, 55(2):174–188, 2002.

[2] A. Blake, C. Rother, M. Brown, and P. Torr. Interactive image segmentation using an adaptive GMMRF model. In *ECCV*, volume 3021, pages 428–441, 2004.

[3] Y. Boykov and G. Funka-Lea. Graph cuts and efficient N-D image segmentation. *IJCV*, 70:109–131, 2006.

[4] Y. Boykov and M. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *ICIP*, pages 105–112, 2001.

[5] Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *ICCV*, pages 26–33, 2003.

[6] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23:1222–1239, 2001.

Figure 11: Uniquely tracking multiple similar, noninteracting players. Each object has its own centroid prediction *(blue dot)* and initialization mask but share the same distance penalty. *(Cropped frames shown.)*

[7] D. Cremers. Dynamical statistical shape priors for level set based tracking. *PAMI*, 28(8):1262–1273, 2006.

[8] D. Cremers, T. Kohlberger, and C. Schnorr. Shape statistics in kernel space for variational image segmentation. *Pattern Recognition*, 36:1929–1943, 2003.

[9] S. Dambreville, Y. Rathi, and A. Tannenbaum. Shape-based approach to robust image segmentation using kernel PCA. In *CVPR*, pages 977–984, 2006.

[10] D. Freedman and T. Zhang. Interactive graph cut based segmentation with shape priors. In *CVPR*, pages 755–762, 2005.

[11] R. Frezza, G. Picci, and S. Soatto. *A Lagrangian formulation of nonholonomic path following*, pages 118–133. The Confluence of Vision and Control. Springer Verlag, 1998.

[12] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *IJCV*, 29(1):5–28, 1998.

[13] P. Kohli and P. Torr. Effciently solving dynamic markov random fields using graph cuts. In *ICCV*, pages 922–929, 2005.

[14] M. Leventon, E. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. In *CVPR*, pages 1316–1324. IEEE, 2000.

[15] E. Maggio and A. Cavallaro. Hybrid particle filter and mean shift tracker with adaptive transition model. In *ICASSP*, pages 221–224, 2005.

[16] J. Malcolm, Y. Rathi, and A. Tannenbaum. A graph cut approach to image segmentation in tensor space. In *Workshop on Component Analysis (CVPR)*, pages 18–25, 2007.

[17] J. Malcolm, Y. Rathi, and A. Tannenbaum. Graph cut segmentation with nonlinear shape priors. In *ICIP*, 2007. (to appear).

[18] Y. Rathi, N. Vaswani, A. Tannenbaum, and A. Yezzi. Particle filtering for geometric active contours with application to tracking moving and deforming objects. In *CVPR*, pages 2–9, 1997.

[19] C. Rother, V. Kolmogorov, and A. Blake. GrabCut: Interactive foreground extraction using iterated graph cuts. In *ACM Trans. on Graphics (SIGGRAPH)*, 2004.

[20] J. Sethian. A fast marching level set method for monotonically advancing fronts. In *Proc. Nat. Acad. Sci.*, volume 93, pages 1591–1595, 1996.

[21] G. Slabaugh and G. Unal. Graph cuts segmentation using an elliptical shape prior. In *ICIP*, pages 1222–5, 2005.

[22] N. Xu, R. Bansal, and N. Ahuja. Object segmentation using graph cuts based active contours. In *CVPR*, pages 46–53, 2003.

[23] L. Yatziv, A. Bartesaghi, and G. Sapiro. O(N) implementation of the fast marching algorithm. *J. of Computational Physics*, 212:393–399, 2006.

[24] X. Zeng, W. Chen, and Q. Peng. Efficiently solving the piecewise constant mumford-shah model using graph cuts. Technical report, Dept. of Computer Science, Zhejiang University, P.R. China, 2006.

Shape Recovery Using Stochastic Heat Flow

Vinay P. Namboodiri and Subhasis Chaudhuri
Department of Electrical Engineering
Indian Institute of Technology, Bombay
Powai, Mumbai 400076, India.
{vinaypn,sc } @ ee.iitb.ac.in

Abstract

We consider the problem of depth estimation from multiple images based on the defocus cue. For a Gaussian defocus blur, the observations can be shown to be the solution of a deterministic but inhomogeneous diffusion process. However, the diffusion process does not sufficiently address the case in which the Gaussian kernel is deformed. This deformation happens due to several factors like self-occlusion, possible aberrations and imperfections in the aperture. These issues can be solved by incorporating a stochastic perturbation into the heat diffusion process. The resultant flow is that of an inhomogeneous heat diffusion perturbed by a stochastic curvature driven motion. The depth in the scene is estimated from the coefficient of the stochastic heat equation without actually knowing the departure from the Gaussian assumption. Further, the proposed method also takes into account the non-convex nature of the diffusion process. The method provides a strong theoretical framework for handling the depth from defocus problem.

1 Introduction

The limited depth of field introduces a defocus blur in images captured with conventional lenses based on the range of depth variation in a scene. This artifact has been used in computer vision for estimating depth in the scene. As discussed in [7], this method of shape recovery is particularly relevant for complex scenes which have a large amount of geometric detail and complex self occlusion relationships which make it difficult to estimate the shape using stereo based methods. In this paper we introduce a new technique for recovering the structure based on the defocus blur. The principal idea that enables our work is that the defocus effect can be modeled in terms of inhomogeneous diffusion (e.g., spatially varying coefficients) of heat using the heat equation. This is because the defocus blur can be modeled as a Gaussian blur, which forms a temporally evolving kernel for the isotropic heat equation. This approach was explored by Favaro *et al.*[4]. Their method had two main shortcomings. First, it could not handle departure from Gaussian assumption in case of self-occlusions. Second, it made an assumption that the diffusion coefficient is a convex function and the solution was based on conjugate gradient based method. In this paper we address both these shortcomings. Here, we propose a model wherein the heat equation is perturbed stochastically. In this approach the departure from the Gaussian blur model is implicitly accounted for in the stochastic perturbation of diffusion. The mathematical existence for the stochastically perturbed heat equation, which is used

here, is analysed by Yip[15] and he has used it to model the dendritic growth of crystal structures. Here we adapt the model for solving the depth from defocus problem by correlating the stochastic heat equation to the defocus blurring process.

The research in depth from defocus was initially introduced by Pentland [12] in which the problem of DFD was posed as an estimation of linear space variant blur. Subsequently, there has been substantial work done using deterministic [6] and statistical techniques [3] in the spatial domain and also by solving in the frequency domain [14]. However, most of the works done assume that the observations do not suffer from self-occlusion. The handling of occlusion effects in depth computation has been addressed in [1],[2],[5]. The extent of departure from the Gaussian shape depends on the nature of depth discontinuity in the scene, which is unknown. Any imperfection in the lens aperture would also change the shape of the blur kernel. Unlike earlier methods, the proposed method can handle such an effect under a unified framework without having to estimate the departure from the assumed model. An interesting recent work has been by Hasinoff and Kutulakos [7], where the authors consider depth from focus as a pixel matching operation. However, this method requires many high resolution images.

2 Defocus as a Stochastically Perturbed Diffusion

In this section we discuss the mathematical basis of stochastic perturbation of the heat equation as a tool to analyse defocused images.

2.1 Defocus as a Diffusion Process

Consider the image formation process in a real aperture camera employing a thin lens [3]. When a point light source is in focus, all light rays that are radiated from the object point and intercepted by the lens converge at a point on the image plane. When the point is not in focus, it is imaged as a circular patch instead of a point. The point spread function (PSF) of the camera describes the image intensity caused by a single point light source. Geometric optics approximates the PSF to a circular disk. However, as discussed in the literature [3],[12] due to diffraction it will be roughly a circular blob with the brightness falling off gradually at the border rather than sharply. The resultant PSF has the general shape of a 2-D Gaussian function [3],[12]. For an equifocal plane the resultant image formed is then given by

$$I(x) = \int f(\tau) h(x,\tau) d\tau, \tag{1}$$

where we adopt x to denote the 2D space co-ordinates in an image, $f(x)$ is the focused image of the scene and h is the space-varying PSF. Here $h(x)$ is given by a circularly symmetric 2-D Gaussian function

$$h(x) = \frac{1}{2\pi\sigma^2} \exp\left(\frac{-x^2}{2\sigma^2}\right), \tag{2}$$

where σ is a function of depth at a given point. It is quite well-known that, for a scene with constant depth the imaging model in eqn(1) can be formulated in terms of the isotropic heat equation [9] given by

$$\frac{\partial u(x;t)}{\partial t} = c\left(\triangle u(x;t)\right), \quad u(x,0) = f(x)$$

where $\triangle u$ is the Laplacian operator. Here the solution $u(x,t)$ taken at a specific time $t = \tau$ plays the role of an image $I(x) = u(x,\tau)$ and $f(x)$ corresponds to the initial condition, i.e. the pin-hole equivalent observation of the scene. Note that we have used $u(x,t)$ to represent the evolution of heat everywhere in the paper. The blurring parameter σ is related to the diffusion coefficient by the following relation [4]

$$\sigma^2 = \frac{2tc}{\gamma} \tag{3}$$

where t is the time variable in the diffusion equation, c is the diffusion coefficient, and γ is a proportionality constant relating the blur radius to the spread (σ) of the blur kernel that can be determined via initial calibration. In the depth from defocus problem, the depth in the scene varies over the image and hence the constant c will actually be $c(x)$, i.e., it will vary over the image. This corresponds to a heat equation in an inhomogeneous medium.

2.2 Stochastic Perturbation

The stochastic form of the isotropic heat equation can be given by

$$du = \alpha(x,t)dW(x,t). \tag{4}$$

where $W(x,t)$ is the brownian motion of a particle ω located at position x and time t. $\alpha(x,t)$ is the diffusion coefficient. The stochastic form of the heat equation corresponds to an Ito diffusion without drift [11].

The addition of stochastic perturbation to the deterministic diffusion equation can be physically thought of as adding thermal fluctuations to the heat diffusion equation. The issues like existence and regularity of the evolution arise by such an addition. These were rigorously studied and proved by Yip[15]. They were studied in the context of crystal growth. However, the same formulation is valid for the defocus problem. The form of the stochastically perturbed diffusion or the stochastic heat equation is given by

$$du = (c(x)\triangle u)dt + \alpha(x,t)dW(t) \tag{5}$$

where $W(t)$ is a spatially correlated infinite dimensional Brownian motion, $dW(t)$ is the Ito's differential and $\triangle u$ corresponds to the Laplacian of u in space. The spatial correlation of W is essential for proving the Gibbs-Thomson condition [15]. This implies that the movement of each particle is not stochastic in space but in time. The Gibbs-Thomson condition is related to the regularity and existence of the solution of eqn(5). Gibbs-Thomson relation is a function which relates the temperature and curvature values in equilibrium for the interface of evolution. Loosely speaking the Gibbs-Thomson condition essentially prescribes an equality between the variation of the energy of the interface and the total divergence of the Gibbs-Thomson relation. These are discussed in detail by Yip in his work[15] where he gives a proof of the Gibbs-Thomson condition for eqn(5).

2.3 Defocusing as a Stochastically Perturbed Diffusion

The defocusing phenomenon has a specific space varying characteristic at surface edges and occluding edges. Consider the particular case as shown in fig. 1. Here we consider the specific case of a surface edge discontinuity which results in self-occlusion. In depth from

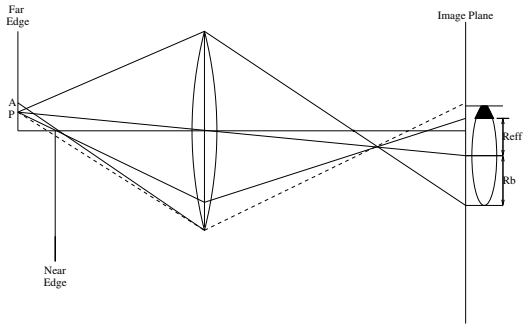


Figure 1: Illustration of the self-occlusion on account of surface discontinuity. For the point P, the point spread function (PSF) is the circular region devoid of the darkened region. For the point A in the scene, the PSF is circular as there is no self-occlusion.

defocus, self occlusion results when a continuum of rays is partially occluded and results in the blur kernel being modified [13]. This is illustrated in fig.1. Here, the rays emanating from the point P are partially blocked due to the surface discontinuity. The image plane is at a distance from the focus point and so the observation of point P results in a blur with radius R_b. However, due to the partial occlusion due to the near edge, the resultant blur instead of being circular is deformed (being R_{eff}). This artifact is present for all points in the observation from the surface edge to the point A. From point A onwards, the blur kernel is unaffected. A similar effect can be observed in the case of an occluding edge as well [1].

There have been a few approaches [1], [2], [5] where the authors have tried to address this problem by explicit modeling of this phenomenon or by adding a post-processing step. However, in our model, due to the stochastically perturbed curvature driven motion along the level sets, it is possible to incorporate this variation implicitly. This is particularly important in correctly estimating the blur kernels along discontinuities like surface edges and occluding edges. This is depicted in fig.1. As shown in the figure, along the surface edge, the contributions from the near and far surface are inhomogeneously mixed and this results in an anisotropic nature to the resultant blur kernel. So, when one does a stochastic curvature driven motion along the level sets, the blur contribution along the surface edge can be appropriately estimated. The non-uniformity of the kernel is implicitly handled in this model. There exists a similar effect when one has an occluding edge as well[1] .

2.4 Evolution Equation

We now proceed to obtain an explicit evolution equation. In order to do this we first obtain an expression for the stochastic perturbation part of eqn(5). Here we consider the recent work done in stochastic level sets [8] and stochastic curvature driven motion [10]. As discussed by Yip[15] , the stochastic perturbation of the eqn(5) can be seen to be given by

$$du(x,t) = \mathbf{n}(x,t)dW(t). \tag{6}$$

where $\mathbf{n}(x,t)$ is the normal to the surface interface $u(t)$ (i.e. the interface $u(x,t)\forall x$). The equivalent deterministic evolution using the level set framework for the geometric heat equation is given by the following equation

$$du(x,t) = \mathbf{n}(x,t)\kappa(x,t) \tag{7}$$

where $\kappa(x,t)$ is the mean curvature of the level set and $\mathbf{n}(x,t)$ is the normal to the level set. Here κ is given by

$$\kappa(x) = \frac{u_{x_1}^2 u_{x_2x_2} - 2u_{x_1}u_{x_2}u_{x_1x_2} + u_{x_1x_1}u_{x_2}^2}{(u_{x_1}^2 + u_{x_2}^2)^{3/2}}, \tag{8}$$

where $x = x_1, x_2$ i.e. two-dimensional space and u_{x_i} refers to $\frac{\partial u}{\partial x_i}, i = 1,2$. The normal $\mathbf{n}(x)$ is given by

$$\mathbf{n}(x) = \frac{\nabla u(x)}{\|\nabla u(x)\|},$$

where $\nabla u(x) = u_{x_1} + u_{x_2}$. The geometric heat equation is similar to the linear heat equation except that it diffuses orthogonal to its gradient and does not diffuse along the direction of the gradient. As a result the stochastic perturbation mainly affects the level set curves and does not affect the homogeneous regions. This is appropriate since any kernel variation for instance due to self occlusion would mainly occur along edges and would be reflected in the stochastic perturbation. The effect of the perturbation is further spread on the homogeneous regions through the deterministic diffusion component.

Now, the stochastic formulation of the above deterministic formulation according to eqn(6) could be written as

$$du(x,t) = \mathbf{n}dW(t), \tag{9}$$

The differential in eqn(9) is the *Ito differential*. This suffers from problems like it is not invariant to the parameterization of the curve, i.e., the evolution depends on the implicit representation of the initial curve and ill posedness, i.e., under certain conditions it approaches the inverse heat equation which is unstable[8],[10]. These difficulties are overcome by introducing the *Stratonovich differential*[11] given by

$$du(x,t) = \mathbf{n} \circ dW(t). \tag{10}$$

The Stratonovich form is in an implicit form and converting it to the explicit Ito form results in an added second order term. This is because of the difference in estimating Ito and Stratonovich differentials. In Ito diffusion the integration happens at the left end point whereas in the Stratonovich case the integration happens at the mid-point while evaluating the integration of the differential[11]. With a single Gaussian perturbation in space, the eqn(10) is written as

$$du(x,t) = \mathbf{n}dW(t) + \frac{1}{2}\triangle u(x,t)\left[\frac{\nabla u(x,t)}{|\nabla u(x,t)|}\right]. \tag{11}$$

The numerical implementation of the scheme for evolution is done by considering a step δt in time and δx in space and is given by[8]

$$u(x,t+\delta t) = u(x,t) + \mathbf{n}\sqrt{\delta t}\ \mathcal{N}_{(0,1)}(t) + \frac{1}{2}\triangle u(x,t)\left[\frac{\nabla u(x,t)}{|\nabla u(x,t)|}\right]. \tag{12}$$

where $\mathscr{N}$ is the noise term and it denotes a standard Gaussian random variable, and the second order term is introduced because of the Stratonovich differential component. This term is a kind of smoothing term and is nothing but the degenerate diffusion component along the edges with the stochastic term corresponding to the diffusion component across the edges. Hence the complete stochastic heat equation would then be

$$u(x,t+\triangle t) = u(x,t) + \mathbf{n}\sqrt{\triangle t}\,\mathscr{N}_{(0,1)}(t) + \frac{1}{2}\triangle u(x,t)\left[\frac{\nabla u(x,t)}{|\nabla u(x,t)|}\right] + c(x)\triangle u(x,t).$$

Since the stochastic perturbation appropriately handles the deformation of the kernel, the diffusion coefficient c is taken to be only a single inhomogeneous coefficient value and not a diffusion tensor.

3 Depth Estimation

We consider the case when we are given two images $I_1(x)$, $I_2(x)$ with different defocus blurs. Then the resultant formulation is

$$\begin{aligned} u(x,t) &= I_1(x) \\ u(x,t+m\delta t) &= I_2(x), \end{aligned} \tag{13}$$

and where the term $u(x,t+m\delta t)$ is obtained from $u(x,t)$ by the evolution in eqn(13) and m is the number of iterations in going from image I_1 to I_2. The evolution equation in eqn(13) blurs the image I_1 with a space-variant blur till it approximates the image I_2 closely enough which is tracked by a discrepancy measure ϕ. The blur parameter σ is related to the diffusion coefficient by the eqn(3) and the blur parameter is directly proportional to the depth in the scene[3] . In order to estimate the depth in the scene one therefore has to estimate the diffusion coefficient for the evolution equation. In a deterministic case one would obtain the following minimization problem:

$$\hat{c}(x) = \arg\min_{c(x)\geq 0} \int\int \phi(u(x,t+dt), I_2(x))dxdt. \tag{14}$$

where $\phi(.)$ is a discrepancy measure and $\hat{c}(x)$ is the diffusion coefficient for the deterministic diffusion equation. However in the stochastically perturbed case, the resultant diffusion coefficient is a combination of deterministic and stochastic diffusion. The deterministic diffusion coefficient is obtained from the contribution from the following part of the evolution equation:

$$du_{\text{det}} = (c_{\text{det}}(x)\triangle u)dt \tag{15}$$

which is the deterministic part of eqn(5). The stochastic diffusion coefficient contribution is obtained by normalizing the stochastic perturbation component in the evolution equation. We recall that the stochastic perturbation component is given by

$$\begin{aligned} du_{\text{st}} &= \mathbf{n}\sqrt{\delta t}\,\mathscr{N}_{(0,1)}(t) + \frac{1}{2}\triangle u(x,t)\left[\frac{\nabla u(x,t)}{|\nabla u(x,t)|}\right] \\ &= \mathbf{n}\circ dW(t) \end{aligned} \tag{16}$$

The stochastic diffusion coefficient is then given by normalizing the stochastic contribution by the corresponding deterministic evolution:

$$c_{\mathrm{st}}(x) = \frac{\mathbf{n} \circ dW(t)}{\kappa \mathbf{n}} \tag{17}$$

where κ is the curvature and $\mathbf{n}$ is the normal.

Thus the combined diffusion coefficient is given by

$$d(x) = c_{\mathrm{det}}(x) + \eta c_{\mathrm{st}}(x) \tag{18}$$

where η is the weight factor which determines the relative weight of the stochastic perturbation. The depth in the scene is obtained by solving for $d(x)$ in a minimization problem of the form

$$\hat{d}(x) = \arg\min_{d(x)\geq 0} \int\int \phi(u(x,t+dt), I_2(x)) dx dt. \tag{19}$$

We adopt a Euclidean distance measure for ϕ. Here the image $I_2(x)$ is assumed to be the more defocused image. However, that may not always be the case, and one can have sections in an image which are more in focus and other sections which are more defocused compared to the corresponding sections in the second image. In that case as an initial step the images are preprocessed and the regions which are more in focus are identified. The diffusion always happens in a forward direction to avoid instabilities that may arise due to backward diffusion. The method used to ensure this is similar the one suggested in [4]. The minimization in eqn.(19) cannot be done using conjugate gradient descent algorithm due to the stochastic perturbation. We adopt a simple simulated annealing scheme to perform the stochastic optimization. The various steps for the algorithm for depth estimation are as follows:

STEP 1: Given the initial images $I_1(x)$, and $I_2(x)$ divide them into sections such that the diffusion is always in the forward direction using the preprocessing step discussed earlier.

STEP 2: Compute u_{n+1} from u_n using the formula for du given in eqn(13).

STEP 3: Compute the discrepancy measure ϕ_n

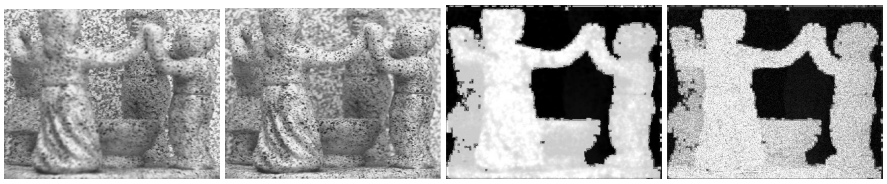
STEP 4: Accept u_{n+1}

- if $\phi_{n+1} < \phi_n$
- otherwise, accept u_{n+1} with probability $\exp\left(\frac{-(\phi_{n+1}-\phi_n)}{T(n)}\right)$.

STEP 5: Loop back to STEP 2 till the stopping criterion is satisfied.

Here $T(n)$ is a time-dependent function that plays the same role as a decreasing temperature. Its choice is crucial. If the temperature decreases too fast the process may get stuck in a local minimum, else if it decreases slowly the convergence is delayed. Here we adopt $T(n) = T_0/\sqrt{n}$ as suggested by Juan *et al.*[8]. The stopping criterion is based on the Euclidean distance measure approaching zero.

The depth estimate is then obtained by considering the deterministic and the stochastic parts separately. For the deterministic part, we assume a constant diffusion coefficient and relate the blur to the time of evolution. The blur cannot be related directly in the stochastic part due to the non-uniform nature of evolution. Hence, in each iteration we normalize



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

Figure 2: Here (a,b) are two real data sets showing the dolls placed at different depths (Images courtesy Favaro [4]). (c) shows the resultant depth map for the deterministic method[4]. (d) shows the corresponding result from the proposed method.

the stochastic perturbation with the corresponding orthogonal diffusion component. We then integrate the corresponding contributions over time to obtain the contribution of the stochastic perturbation to the blurring process. The final depth estimate is obtained as the joint contribution of the deterministic and stochastic components.

The depth obtained in this method has a space-varying characteristic, i.e., the problem solved is equivalent to space varying point spread function (PSF) estimation. Further due to the stochastic nature, the self occlusion effects and other imperfections are implicitly handled by the method when it does a stochastic perturbation of the blur model.

4 Experiments

The algorithm has been tested with real images and the results are compared with state of the art techniques. The method works quite well on all these test data sets.

The experimental setup shown in fig.2 is the "dolls" data set[4]. The images were taken with varying lens to image plane distances to obtain different amount of defocus in different observations. The Fig.2(c) shows the depth map estimated by the deterministic method[4] and Fig.2(d) shows the depth map obtained by the proposed method. Once again we can clearly identify the depth boundaries from the recovered depth map, justifying the usefulness of the proposed algorithm. The different dolls are clearly visible to be at different depths.

A challenging data set is the "hair" data set used in [7]. The data set is of a wig with a messy hairstyle surrounded by several artificial plants. This data poses challenging self occlusion and complex structure issues. Fig. 3(a,b) shows the 2 input images used. Fig. 3(c) shows the depth map obtained for the deterministic method [4]. As can be seen, the method does not handle the self occlusion and non-convex diffusion coefficient issues efficiently. Fig. 3(d) shows the depth map obtained from the confocal stereo method [7]. However, they have images from 13 aperture settings each with 61 focal settings. Fig. 3(e) shows the depth map obtained from the proposed method using two input images which is comparable to the depth map in [7] from many images. The results are illustrated clearly and additional results are provided at *http://vinaypn.googlepages.com/stochdfd*.

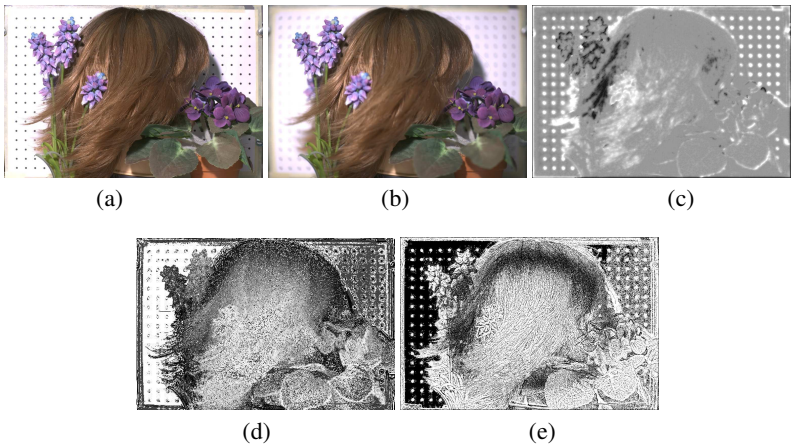


Figure 3: Here (a,b) are two real data sets showing a wig and flowers (Images courtesy [7]). (c) shows the resultant depth map for the deterministic method[4]. (d) shows the corresponding result from [7] (which uses images from 13 aperture, each with 61 focus settings) and (e) depicts the result from the proposed method (using only 2 images).

5 Conclusion

In this paper we have proposed a method based on stochastic perturbation of diffusion for solving the depth from defocus problem. The main contribution here has been in incorporating a stochastic formulation of the blur model which can effectively handle variations in the blur from the standard Gaussian blur model. The variations arise in the real world due to aberrations in the lenses and aperture and are experimentally too elaborate to measure. Further the problem of deformation of the Gaussian kernel due to self occlusion is also implicitly handled. We demonstrate that improved results can be obtained using the proposed technique. The proposed method also takes into account the non-convex nature of diffusion propagation.

It may be noted that most researchers in the area of structure recovery have pointed out the need for regularization of the recovered surface. The proposed method does not impose any such constraint while recoverying the depth. Currently we are investigating the possibility of incorporating a spatial smoothness constraint during diffusion propagation to improve accuracy.

6 Acknowledgments

Financial assistantship under the *Swarnajayanti* fellowship scheme of Department of Science and Technology, India is gratefully acknowledged.

References

[1] N. Asada, H. Fujiwara, and T. Matsuyama. Seeing behind the scene: Analysis of photometric properties of occluding edges by the reversed projection blurring model.

IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(2):155–167, February 1998.

[2] S.S. Bhasin and S. Chaudhuri. Depth from defocus in presence of partial self occlusion. In *Proc. Eighth IEEE Int'l Conf. on Computer Vision*, volume 1, pages 488–493, July 2001. Vancouver, Canada.

[3] S. Chaudhuri and A. N. Rajagopalan. *Depth From Defocus: A Real Aperture Imaging Approach*. Springer Verlag, New York, 1999.

[4] P. Favaro, S. Osher, S. Soatto, and L. Vese. 3d shape from anisotropic diffusion. In *Proceedings of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 179–186, 2003. Madison, Wisconsin, USA.

[5] P. Favaro and S. Soatto. Seeing beyond occlusions (and other marvels of a finite lens aperture)learning shape from defocus. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR 03)*, volume 2, pages 579–586, 2003. Madison, Wisconsin, USA.

[6] Paolo Favaro and Stefano Soatto. *3-D Shape Estimation and Image Restoration: Exploiting Defocus and Motion-Blur*. Springer-Verlag, London, 2007.

[7] S.W. Hasinoff and K.N. Kutulakos. Confocal stereo. In *Proc. Ninth European Conference on Computer Vision*, pages 620–634, May 2006. Graz, Austria.

[8] O. Juan, R. Keriven, and G. Postelnicu. Stochastic motion and the level set method in computer vision: Stochastic active contours. *Internation Journal of Computer Vision*, 69(1):7–25, August 2006.

[9] J. J. Koenderink and A. J. van Doorn. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.

[10] P.-L. Lions and Panagiotis E. Souganidis. Fully nonlinear stochastic partial differential equations. *C. R. Acad. Sci. Paris, t. 331, Srie I*, pages 1085–1092, 1998.

[11] B. Oksendal. *Stochastic Differential Equations: An Introduction with Applications*. Springer Verlag, New York, sixth edition, 2004.

[12] A. P. Pentland. A new sense for depth of field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4):523–531, July 1987.

[13] Y. Y. Schechner and N. Kiryati. Depth from defocus vs. stereo: How different really are they? *International Journal of Computer Vision*, 39(2):141–162, September 2000.

[14] Y. Xiong and S.A. Shafer. Moment filters for high precision computation of focus and stereo. In *Proc. Intl Conf. Intelligent Robots and Systems*, pages 108–113, August 1995. Pittsburgh, Pennsylvania, USA.

[15] N.K. Yip. Existence of dendritic crystal growth with stochastic perturbations. *Journal of Nonlinear Science*, 8:491–579, 1998.

Integrating Stereo with Shape-from-Shading derived Orientation Information

Tom S. F. Haines and Richard C. Wilson

Department of Computer Science
University of York
Heslington, York, UK

thaines,wilson@cs.york.ac.uk

Abstract

Binocular stereo has been extensively studied for extracting the shape of a scene. The challenge is in matching features between two images of a scene; this is the correspondence problem. Shape from shading (SfS) is another method of extracting shape. This models the interaction of light with the scene surface(s) for a single image. These two methods are very different; stereo uses surface features to deliver a depth-map, SfS uses shading, albedo and lighting information to infer the differential of the depth-map.

In this paper we develop a framework for the integration of both depth and orientation information. Dedicated algorithms are used for initial estimates. A Gaussian-Markov random field then represents the depth-map, Gaussian belief propagation is used to approximate the MAP estimate of the depth-map. Integrating information from both stereo correspondences and surface normals allows fine surface details to be estimated.

1 Introduction

Binocular stereo is a long-standing problem in computer vision[17]. It enables the construction of 3D models from two 2D images, by solving the correspondence problem, where matching features are found between two images such that the matched features are at the same location on the object. If camera calibration is then available depth can be reconstructed from such matches. A standard preprocessing step involves rectifying the images, so that they represent the images taken from an idealised horizontal parallel camera pair. Given a rectified image pair features can only match features on the same scan-line. In the dense stereo problem where every pixel is a feature a disparity map is created, where the disparity assigned to each pixel is the offset along the x-axis to its matching pixel in the other image.

Dense stereo algorithms may be divided into two steps. First is the calculation of a matching cost for each disparity at each location, represented by the $3D$ Disparity Space Image (DSI). In areas with strong cues a DSI gives a clear indication of actual disparity, but in relatively uniform areas it will not distinguish the correct disparity from incorrect disparities. Suggested approaches include normalised cross-correlation[17] and sliding

windows[3]. A common choice is to use individual pixel dissimilarities and rely on modelling assumptions in the second step to recover a reasonable solution.

The second step is the selection of disparities to find a consistent solution. Simply selecting the best matching cost does not work well because of noise and ambiguities, so there is a need for modelling assumptions. Modern approaches use techniques such as dynamic programming[1], graph cuts[4] and belief propagation[21]. The most common modelling assumption is a smoothness term, which is often equivalent to assuming piecewise planar surfaces or fronto-parallel piecewise planar surfaces. Whilst these approaches do well in regions with strong stereo cues uniform areas are generally plane fitted or interpolated, regardless of actual shape.

Shape from Shading (SfS) relies on the shading information available from a single image. It is premised on the intensity of light reflected by a surface being related to the angle between the surface and light source(s). It therefore provides information about surface gradient. The approach was pioneered by Horn[11] and then Ikeuchi and Horn[12]. Whilst complete orientation is desired surface intensity only provides tilt information. Again, as with stereo, modelling assumptions are used to resolve the ambiguities. A smooth surface assumption is again common, though this is smoothing surface orientation rather than disparity. SfS algorithms generally assume Lambertian surface reflectance and constant albedo over the surface in question, and therefore need constant albedo objects.

Stereo algorithms do not perform effectively in areas of uniform texture. Such regions will generally either be interpolated or plane fitted, which is not necessarily a true reflection of the surface shape. In contrast SfS can operate only in areas where albedo can be inferred, so a uniform albedo assumption needs to be used. This makes SfS ideal for filling in areas where stereo has insufficient information[14]. Furthermore, stereo provides information about albedo, which is necessary for SfS. In combining these ideas we have an improved set of modelling assumptions allowing for a surface estimate with greater detail. Leclerc and Bobick[14] have used stereo to provide initialisation and boundary constraints for SfS. Cryer, Tsai and Shah[7] combine SfS and stereo in the frequency domain, using a high pass filter for the SFS and a low pass filter for the stereo. Other work has taken an object centred approach[10, 16]. Here a model is initialised with a stereo algorithm and then optimised to fit both stereo and shading information. The method needs a good initialisation however, and is not effective with only a single stereo pair for initialisation. Jin, Yezzi and Soatto[13] assume the image is divided into areas of texture and constant albedo and apply separate cost functions to each area and solve with level sets. Shao et al[18] use an additional cost for the difference between SfS irradiance in the left image and image irradiance at the corresponding point in the right image. This combines depth and shading into a single cost function.

The paper is organised as follows; section 2 gives the problem formulations for stereo and SfS. Section 3 gives the core details of our algorithm. Section 4 describes albedo estimation whilst sections 5 and 6 detail the SfS and stereo algorithms respectively. Section 7 presents results and, finally, section 8 concludes.

2 Problem Formulation

The goal is to use the methods of stereo and shape from shading in a complementary way. Here we briefly describe the problem formulations of both.

2.1 Stereo

The images captured by the camera pair are initially rectified using the camera calibration before processing. The input to the stereo problem is therefore a rectified image pair, the images notionally referred to as the left, $I_L(x,y)$ and right, $I_R(x,y)$. The output is a disparity map, $D(x,y)$, representing the dense correspondences between images. Rectification has ensured that epipolar line are horizontal, therefore disparities are offsets on the x-axis, i.e. $I_L(x,y)$ corresponds to $I_R(x+D(x,y),y)$. The process may be divided into two steps, first a $DSI(x,y,d)$ is defined expressing the cost of matching $I_L(x,y)$ and $I_R(x+d,y)$. Modelling assumptions are then used to select an optimal set of matches which produces the solution, $D(x,y)$. Given camera calibration disparity may be converted into a depth map.

2.2 Shape from Shading

SfS uses the image intensity of a single image[1], $L(x,y)$. The goal of SfS is to recover surface orientation for each pixel, $\mathbf{n}(x,y)$. Under a single light source and Lambertian reflectance model the surface normals are related to the image intensity via

$$L(x,y) = A(x,y)(\mathbf{n}(x,y)\cdot\mathbf{s}) \tag{1}$$

where $\mathbf{s}$ is the light-source direction and $A(x,y)$ is the apparent albedo at (x,y) in the image. The goal of SfS is to recover the surface normals given the luminance map, albedo map and the light source direction. As equation 1 only constrains the angle between the light source and surface normal modelling assumptions such as surface smoothness must be introduced to solve the problem. In principle depth can be recovered from the normal map by integrating over the surface. This is neither straightforward nor accurate however.

3 Integrating Depth and Orientation Information

Once we have a field of surface normals to hand we can use it to provide information about scene shape by integration. Traditionally this is done using a global integration method, such as that of Frankot and Chellapa[9]. Depth information is also available however, provided directly by the stereo algorithm. Our goal is then to combine these two sources of information to produce an improved estimate of the surface. We do this within the framework of Gaussian belief propagation. This enables us to define the required surface as the MAP estimate of a Markov random field, and to combine the two sources of information in a probabilistic way.

3.1 Belief Propagation

Belief propagation is a powerful method for finding the posterior distribution of a Markov random field. It has previously been used with discrete distributions to find stereo disparities[8]. In our case we have to recover disparity to sub-pixel accuracy otherwise surface normals will not provide much information, i.e. the change in orientation produced by a unit change in disparity is often an order of magnitude more than the SfS derived orientation information can provide. Using discrete distributions would result in an infeasibly large

[1] We use the luminance channel of the *Luv* colour space for the intensity. Experimentation has shown this to be in reasonable agreement with Lamberts law for non-specular objects with the cameras used. There is an implicit white light assumption being made here.

number of disparity labels. This makes it essential to use continuous density functions representing continuous disparity measurements. One tractable solution is to use Gaussian distributions. The beliefs are then defined by the mean and variance of the Gaussian distribution, allowing orientation information to be effectively used. We adopt this approach in this paper.

Loopy belief propagation works by iteratively passing messages between nodes of the MRF. The message that a node t passes to its neighbour s is[19]

$$m_{ts}^{(n)}(x_s) = \alpha \int_{x_t} \psi_{st}(x_s, x_t)\psi_t(x_t, y_t) \prod_{u \in N/s} m_{ut}^{(n-1)}(x_t)dt \tag{2}$$

Here x_t is the disparity at node t; $\psi_{st}(x_s, x_t)$ is the compatibility distribution between the disparities at t and s; $\psi_t(x_t, y_t)$ is the distribution of disparities inferred from the observed evidence y_t; $m_{ut}^{(n-1)}(x_t)$ is a message from the previous iteration; and the set N/s is the neighbourhood of t excluding s. We can then compute the belief at node t using

$$b_t^{(n)} = \alpha \psi_t(x_t, y_t) \prod_{u \in N} m_{ut}(x_t) \tag{3}$$

We adopt a variant of the Gaussian algebra of Cowell[6]. The Normal distribution is defined as a function of the precision $\mathbf{P}$ and the precision times the mean $\mathbf{P}\mu$, which we will refer to as the p-mean. The precision is equal to the inverse covariance matrix, i.e. $\mathbf{P} = \Sigma^{-1}$. We have

$$\phi[\mathbf{P}\mu, \mathbf{P}] = \alpha \exp\left[-\frac{1}{2}(\mathbf{x} - \mu)^T \mathbf{P}(\mathbf{x} - \mu)\right] \tag{4}$$

The reason for defining ϕ in this way is that it produces a simple set of rules for manipulating the distributions, which are given in Appendix A.

Under our Gaussian model the stereo algorithm is used to give an initial estimate of the disparities. At a point t in the image the stereo algorithm gives a set of measurements, y_t, which are used to infer a distribution for the disparities, x_t. This is modelled by a Normal distribution, $\psi_t(x_t, y_t) = \phi[\mathbf{P}_t\mu_t, \mathbf{P}_t]$. The mean μ and precision $\mathbf{P}$ for this distribution are computed from the stereo algorithm as detailed in section 6.

The compatibility distribution between two neighbouring points in the image s and t is also modelled by a normal distribution. If the disparity at t is x_t then we would expect the disparity at s to be $x_t + z_{ts}$ where z_{ts} is the disparity change predicted by integrating the surface normals along the path from t to s. The compatibility distribution $\psi_{st}(x_s, x_t)$ is therefore defined as a Normal distribution with mean $x_t + z_{ts}$ and a fixed precision P_n which reflects the accuracy of the surface normals. We therefore obtain

$$\psi_{st}(x_s, x_t) = \phi\left[P_n \begin{pmatrix} -z_{ts} \\ z_{ts} \end{pmatrix}, P_n \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}\right] \tag{5}$$

Since the points are neighbours in the image we can assume that the surface normal direction is constant along the path between them, and use an interpolated surface orientation at the half way point. This is in fact necessary to avoid bias in the result. The two separate processes therefore influence the MRF in different ways; the local measurement process models the depth information and the compatibility between sites is used to incorporate the orientation information.

Since the distributions are Normal, the messages are also Normal distributions. We begin by defining the following quantities:

$$P_0 = P_t + \sum_{u \in N/s} P_{ut} \qquad P_0\mu_0 = P_t\mu_t + \sum_{u \in N/s} P_{ut}\mu_{ut}$$

These are the local precision and p-mean respectively, excluding the message we are currently computing. The new message $m_{ts}^{(n)}(x_s)$ is a Normal distribution which we will define as

$$m_{ts}^{(n)}(x_s) = \phi[\mathbf{P}_{ts}\mu_{ts}, \mathbf{P}_{ts}]$$

Applying Eqn. 2, we obtain the update rules:

$$\begin{aligned} P_{ts} &\leftarrow P_n - P_n(P_n + P_0)^{-1}P_n \\ P_{ts}\mu_{ts} &\leftarrow P_n z_{ts} + P_n(P_n + P_0)^{-1}(P_0\mu_0 - P_n z_{ts}) \end{aligned} \tag{6}$$

We iteratively apply these rules to find an estimate of the MAP disparity map. The beliefs are given by

$$b_t^{(n)} = \alpha\psi_t(x_t, y_t)\prod_{u \in N} m_{ut}(x_t) = \phi\left[P_t\mu_t + \sum_{u \in N} P_{ut}\mu_{ut}, P_t + \sum_{u \in N} P_{ut}\right]$$

so the mean, and hence the estimated disparity, is

$$\mu(t) = (P_t\mu_t + \sum_{u \in N} P_{ut}\mu_{ut})(P_t + \sum_{u \in N} P_{ut})^{-1} \tag{7}$$

4 Albedo Estimation

Under the Lambertian reflectance assumption SfS requires an albedo map as input. The *surface texture* consists of albedo and colour; colour is taken to be the (u,v) channels of *Luv* colour space. For an arbitrary texture it is impossible to distinguish texture variation from shading; this is the basis of '3*D*' effects in user interfaces. It has already been noted that in variable texture regions stereo matching is effective, so additional SFS information is only necessary in relatively uniform regions. Uniform regions allow us to ignore texture variation.

We begin by segmenting the image into uniform regions with mean shift[5] to obtain a set of regions, R. Within each of these regions the colour is uniform and the albedo is assumed to be uniform. The luminance L will however vary across the region because of shading effects. In order to correctly compute the albedo of a region we need to account for shading effects using Equation 1. Given a field of surface normals $\mathbf{n}(x,y)$ we can estimate the albedo at each pixel via the relation $A(x,y) = L/(\mathbf{n}(x,y)\cdot\mathbf{s})$. For individual pixels this is not reliable due to inaccurate normal estimation. As albedo is constant an accurate estimate can be obtained by averaging over each region

$$A_r = \frac{1}{|r|}\sum_{(x,y)\in r}\frac{L(x,y)}{(\mathbf{n}(x,y)\cdot\mathbf{s})} \tag{8}$$

where $r \in R$. This requires a field of surface normals; as it is reasonably robust to noise this may be obtained directly from the stereo algorithm.

	Boot Strap	Smoothed Boot Strap	Our Algorithm
Frame	1.62(13.5%)	1.22(2.2%)	1.08(1.0%)
Head	1.84(13.7%)	1.55(0.2%)	1.90(1.9%)
Head (centre)	1.73(10.0%)	1.47(0.1%)	1.40(0%)

Table 1: Statistics, see text for details.

5 Shape from Shading Algorithm

Equation 1 only constrains the angle between the surface normal and light source, i.e. surface normals must lie on a cone whose angle is defined by the ratio L/A. To remove this ambiguity we introduce two constraints, which are a) that surface normals should vary smoothly across the surface, and b) at occluding boundaries, the surface normals lie in the image plane and point away from the boundary. We adopt the Worthington and Hancock[20] algorithm to solve for the field of surface normals by alternately smoothing and re-projecting onto the cone. Applying this method gives us a fields of surface normals for either image, the framework only uses the left images orientation information however.

6 Stereo Algorithm

The DSI of a single pixel can not be accurately represented with a single Gaussian. A disparity value and its confidence can be however, hence the need for a stereo algorithm to select a *reasonable* disparity for each pixel. A modified version of the algorithm of Meerbergen et al.[15] is used. The modification is such that, in addition to the best disparity, it also outputs all other disparities within a given tolerance of the best, as an indication of confidence. It uses the Birchfield and Tomasi's[2] sampling invariant dissimilarity measure, the resulting disparities can therefore be considered as ranges, ± 0.5 the given value, rather than as infinitesimal points. A Gaussian can therefore be fitted to each pixels set of disparities for use by the Gaussian belief propagation step. Occluded pixels with no disparities are assigned an evidence of $\psi_t(x_t, y_t) = \phi[0,0]$.

Both the SfS initialisation and albedo estimation steps require surface normals to be extracted from the initial disparity map, this may be done with camera calibration information. Directly estimating surface normals by differentiating a *discrete* depth-map does not work however. Therefore the belief propagation process is run to obtain an initial smooth surface; for this first run orientation is provided by plane fitting the uniform colour segments. To further reduce noise least squares planes are fitted to a 11×11 window around each normal and the plane perpendiculars used; this is necessary to obtain a robust albedo estimate.

7 Experimental Results

We have evaluated our method using a number of stereo pairs with ground truth data. Standard stereo tests[17] are not fit for our purposes, in part because they do not match the single known light source requirement but also because they give ground truth in terms of either discrete disparities or fitted planes. As we obtain surfaces to a much finer resolution we need ground truth data with disparity maps at sub-pixel resolution.

Using a Cyberware 3030 head scanner and two Canon S70s in the standard stereo position a data set with ground truth data has been captured. It was calibrated both before and after the capture session with a 3D calibration target. The stereo pair backgrounds are

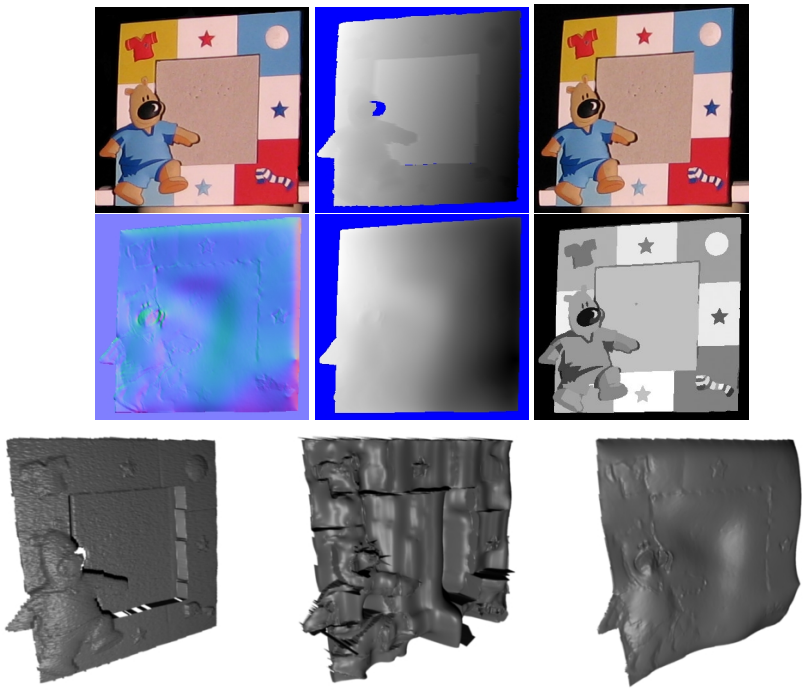


Figure 1: Results for a photo frame, see text for details.

masked out; the ground truth disparity map for the human head is masked out in problem areas, such as eyes and hair.

We illustrate the algorithm with the picture frame given in figure 1 and the head given in figure 2. The figures are arranged as left image, ground truth disparity then right image on the first row, output orientation map, disparity map and albedo map on the second row. The final row contains renders of the 3D models, first the ground truth, then the smoothed boot strap and finally the output. The frame is smoothed considerably by the algorithm. Whilst some overall structure has been lost details not visible in the initialisation are apparent, primarily the decoration on the frame. It additionally shows the effectiveness of the albedo estimation. The algorithm produces a reasonable visual result for the head, unlike the bootstrap algoirthm. Again, it captures details not visible in the bootstrap.

Table 1 compares the algorithm statistically, with the boot strap algorithm[15] in the first column then the smoothed version used for albedo estimation followed by the final result. We provide two values in each case. The first is the average disparity difference from ground truth for pixels classified as inliers, the second is in brackets and is the percentage of outliers. We define inliers as disparity values within 8 pixels of the ground truth disparity. For the frame the results are clear cut, but for the head the numbers indicate that our algorithm has made it worse, though the renders indicate otherwise. For the smoothed version the error is equally distributed, but for the output from our algorithm the error is primarily in the ears and edges of the face. This is because the lambertian shading model is insufficient in these regions. The *head (centre)* row shows the statistics

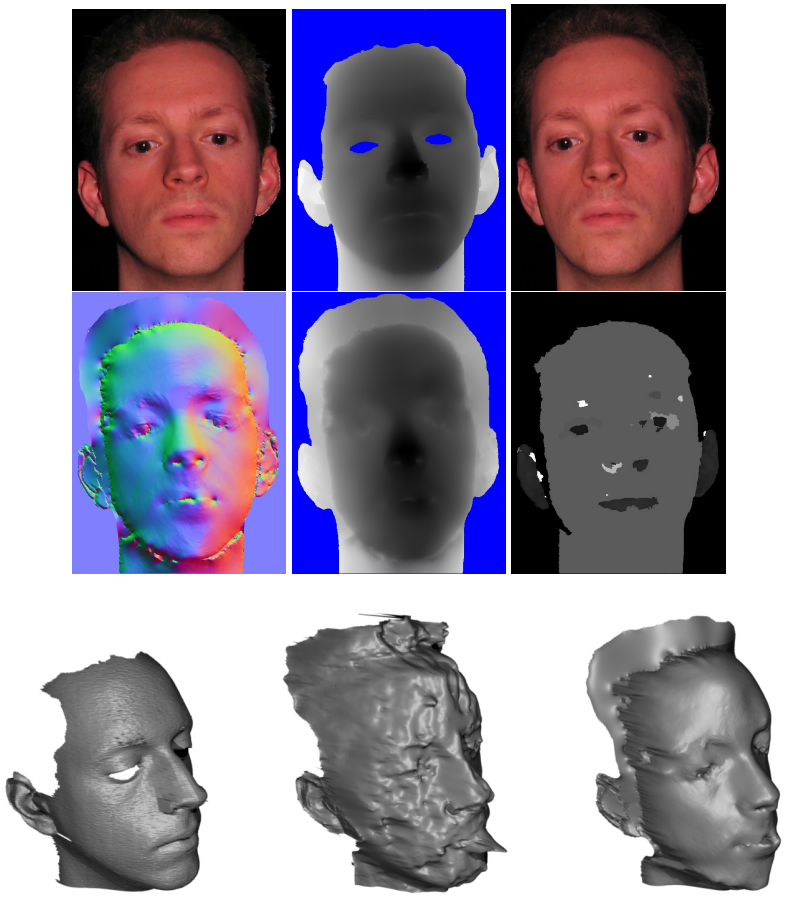


Figure 2: Results for a head, see text for details.

when the ears and edges of the head are masked out.

8 Conclusions

We have presented a method of integrating shape from shading information with stereo information using Gaussian belief propagation. This method efficiently delivers a continuous estimate of disparity and is relatively easy to implement. Our results show an improvement in the fine surface details when shading information is used, leading to more accurate and visually pleasing models.

Much possible future work exists in this area. The greatest weakness of this approach is SfS requiring a single known light source. Two future directions may be found in using another source of orientation information or removing this constraint from SfS, with light source estimation and support for multiple light sources.

A Gaussian Belief Propagation

Multiplying, we get

$$\phi[\mathbf{P_1}\mu_1,\mathbf{P_1}]\phi[\mathbf{P_2}\mu_2,\mathbf{P_2}] = \phi[\mathbf{P_1}\mu_1+\mathbf{P_2}\mu_2,\mathbf{P_1}+\mathbf{P}_2] \tag{9}$$

If we add an additional independent variable, we get

$$\text{Ext}(\phi[\mathbf{P}\mu,\mathbf{P}] = \phi\left[\begin{pmatrix}\mathbf{P}\mu\\0\end{pmatrix},\begin{pmatrix}\mathbf{P}&0\\0&0\end{pmatrix}\right] \tag{10}$$

Finally, if we marginalise over the first variable, we get

$$\text{Marg}_1(\phi[\mathbf{P}\mu,\mathbf{P}]) = \phi[\mathbf{h}_2-\mathbf{P}_{12}\mathbf{P}_{11}^{-1}\mathbf{h}_1,\mathbf{P}_{22}-\mathbf{P}_{12}\mathbf{P}_{11}^{-1}\mathbf{P}_{12}^T] \tag{11}$$

where $\mathbf{P} = \begin{pmatrix}\mathbf{P}_{11}&\mathbf{P}_{12}\\\mathbf{P}_{12}^T&\mathbf{P}_{22}\end{pmatrix}$ and $\mathbf{P}\mu = \begin{pmatrix}\mathbf{h}_1\\\mathbf{h}_2\end{pmatrix}$

Combining the local distribution with previous messages:

$$\begin{aligned}\psi_t(x_t,y_t)\prod_{u\in N/s}m_{ut}^{(n-1)}(x_t) &= \phi[P_{tt}\mu_{tt},P_{tt}]\prod_{u\in N/s}\phi[P_{ut}\mu_{ut},P_{ut}]\\ &= \phi[P_{tt}\mu_{tt}+\sum_u P_{ut}\mu_{ut},P_{tt}+\sum_u P_{ut}]\\ &= \phi[P_0\mu_0,P_0]\end{aligned} \tag{12}$$

Extending the distribution to incorporate x_s, we get

$$\text{Ext}(\phi[P_0\mu_0,P_0]) = \phi\left[\begin{pmatrix}P_0\mu_0\\0\end{pmatrix},\begin{pmatrix}P_0&0\\0&0\end{pmatrix}\right]$$

Then combining with $\psi_{st}(x_s,x_t)$:

$$\psi_{st}(x_s,x_t)\text{Ext}(\phi[P_0\mu_0,P_0]) = \phi\left[\begin{pmatrix}P_0\mu_0-P_nz_{ts}\\P_nz_{ts}\end{pmatrix},\begin{pmatrix}P_n+P_0&-P_n\\-P_n&P_n\end{pmatrix}\right]$$

Finally, we marginalise to find the new message

$$\begin{aligned}m_{ts}^{(n)}(x_s) &= \alpha\text{Marg}_1(\phi\left[\begin{pmatrix}P_0\mu_0-P_nz_{ts}\\P_nz_{ts}\end{pmatrix},\begin{pmatrix}P_n+P_0&-P_n\\-P_n&P_n\end{pmatrix}\right])\\ &= \phi[P_nz_{ts}+P_n(P_n+P_0)^{-1}(P_0\mu_0-P_nz_{ts}),P_n-P_n(P_n+P_0)^{-1}P_n]\end{aligned}$$

so the message update rules are

$$\begin{aligned}P_{ts} &\leftarrow P_n-P_n(P_n+P_0)^{-1}P_n\\ P_{ts}\mu_{ts} &\leftarrow P_nz_{ts}+P_n(P_n+P_0)^{-1}(P_0\mu_0-P_nz_{ts})\end{aligned} \tag{13}$$

References

[1] A. A. Amini, T. E. Weymouth, and R. C. Jain. Using dynamic programming for solving variational problems in vision. *Pattern Analysis and Machine Intelligence*, 12(9):955–867, 1990.

[2] S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *Pattern Analysis and Machine Intelligence*, Vol. 20, No. 4:401–406, 1998.

[3] A. F. Bobick and S. S. Intille. Large occlusion stereo. *International Journal of Computer Vision*, 33(3):181–200, 1999.

[4] Y.i Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[5] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence*, Vol. 24, No. 5:603–619, 2002.

[6] R. Cowell. *Learning in Graphical Models*, chapter Advanced Inference in Bayesian Networks. MIT Press, 1998.

[7] J. E. Cryer, P. S. Tsai, and M. Shah. Integration of shape from shading and stereo. *Pattern recognition*, 28(7):1033–1043, 1995.

[8] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *IEEE CVPR*, 1:261–268, 2004.

[9] R.T. Frankot and R. Chellappa. A method for enforcing integrability in shape from shading algorithms. *Pattern Analysis and Machine Intelligence*, 10:439–451, 1988.

[10] P. Fua and Y. G. Leclerc. Object-centered surface reconstruction: combining multi-image stereo and shading. *International Journal of Computer Vision*, 16(1):35–56, 1995.

[11] B. K. P. Horn. *The Psychology of Computer Vision*, chapter Obtaining shape from shading information, pages 115–155. McGraw Hill, 1975.

[12] K. Ikeuchi and B.K.P. Horn. Numerical shape from shading and occluding boundaries. *Artificial Intelligence*, 17(3):141–184, 1981.

[13] H. Jin, A. Yezzi, and S. Soatto. Stereoscopic shading: Integrating multiframe shape cues in a variational framework. In *CVPR*, volume 1, pages 169–176, 2000.

[14] Y. G Leclerc and A. F. Bobick. The direct computation of height from shading. In *CVPR*, pages 552–558, 1991.

[15] G. Van Meerbergen, M. Vergauwen, M. Pollefeys, and L. Van Gool. A hierarchical symmetric stereo algorithm using dynamic programming. *International Journal of Computer Vision*, Vol. 47:275–285, 2002.

[16] D. Samaras, D. Metaxas, P. Fua, and Y. G. Leclerc. Variable albedo surface reconstruction from stereo and shape from shading. In *CVPR*, volume 1, pages 480–487, 2000.

[17] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1):7–42, 2002.

[18] M. Shao, R. Chellappa, and T. Simchony. Reconstructing a 3-d depth map from one or more images. *CVGIP: Image Understanding*, 53(2):219–226, 1991.

[19] Y. Weiss and W. T. Freeman. Correctness of belief propagation in gaussian graphical models of arbitrary topology. *Neural Computation*, 13(10):2173–2200, 2001.

[20] P.L. Worthington and E.R. Hancock. New constraints on data-closeness and needle map consistency for shape-from-shading. *Pattern Analysis and Machine Intelligence*, 21(12):1250–1267, 1999.

[21] J.S. Yedidia, W.T. Freeman, and Y. Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *Information Theory*, 51(7):2282–2312, 2005.

Active Segmentation and Adaptive Tracking Using Level Sets

Zezhi Chen[1] and Andrew M Wallace[2]
[1]School of Mathematics and Computer Sciences
[2]School of Engineering and Physical Sciences
Joint Research Institute in Signal and Image Processing
Heriot-Watt University, Edinburgh, UK, EH14 4AS
{Zc19, A.M.Wallace}@hw.ac.uk

Abstract

We describe algorithms for active segmentation (AS) of the first frame, and subsequent, adaptive object tracking through succeeding frames, in a video sequence. Object boundaries that include different known colours are segmented against complex backgrounds; it is not necessary for the object to be homogeneous. As the object moves, we develop a tracking algorithm that adaptively changes the colour space model (*CSM*) according to measures of similarity between object and background. We employ a kernel weighted by the normalized Chamfer distance transform, that changes shape according to a level set definition, to correspond to changes in the perceived 2D contour as the object rotates or deforms. This improves target representation and localisation. Experiments conducted on various synthetic and real colour images illustrate the segmentation and tracking capability and versatility of the algorithm in comparison with results using previously published methods.

1 Introduction

In this paper, we address the problem of segmentation and tracking of human subjects through video sequences, in which the subject is defined by an enclosing contour and a colour distribution within that contour, and the background may be static (fixed camera) or moving (panning camera) and defined by another colour distribution. In general, the colours within the foreground and background may change due to a different viewpoint or change of illumination. The work is founded on earlier work on mean-shift [5], level-set [2][7] and combined [3]methods to segment images and track deformable shapes in video sequences. In summary, there are three improvements over previous work.

The first process is segmentation on the first frame of the sequence to define the shape to be tracked. This uses an active segmentation (AS) algorithm based on level set methods and a multi-phase colour model. However, we have defined a general variational formulation which combines the Minkowski distance L_2 and L_3 of each channel and their homogenous regions in the index, as a change to the previous CVV model [1]. This method finds whole object boundaries that include different known colours, even in very

complex background situations, and shows improvement in synthetic data, in which the additive noise is non-Gaussian and asymmetric, and on real image data.

Second, we have developed an adaptive object tracking algorithm that combines AS and a mean shift tracking. The tracking algorithm has two phases. Assuming a current shape in a frame of index, *i*, then the mean shift algorithm can be used to find the most likely position of that same shape in frame $i+1$. Then, the AS algorithm deforms the contour to find a contour that better fits the data in the same $(i+1)^{th}$ frame. The approach is adaptive, in that it allows both deformation of the contour, and a change of the colour space model (*CSM*), the latter building on the work by Collins et al. [4] throughout the processing of a video. However, we sort the different *CSM*s using the Bhattacharyya coefficient which is an approximate measurement of the amount of overlap between the two distributions of foreground and background, instead of using the variance ratio measure of the distribution of likelihood values.

The third modification, when we obtain the boundary of a tracked object, is to use a kernel weighted by the normalized chamfer distance transform to improve the accuracy of target representation and localization. This replaces the more usual Epanechnikov kernel[6]. Comparative experiments show that our approach is more successful in tracking the object through video sequences, as both foreground and background colour distributions are better matched to the separated regions within the data.

2 Segmentation by Level Sets

2.1 Description of the Model

The basic idea in active contour segmentation is to evolve a curve, subject to constraints, in order to detect objects in the image. *"Let Ω be a bounded open subset of $\mathbf{R^2}$, with $\partial\Omega$ the boundary. Let $\boldsymbol{I}$ be a given image such that $\boldsymbol{I} : \overline{\Omega} \longrightarrow \mathbf{R}$. Let C(s) : [0, 1] $\longrightarrow R^2$ be a piecewise parameterized C^1 curve"* [1]. We make the following assumptions: 1) **I** is composed by a maximum of **M** regions Ω_i; 2) the interface between the regions $\partial\Omega$ is regular. Our method also includes the minimization of an energy based function to perform segmentation. Describing image segmentation by a variational model increases the flexibility of the representation, allowing the future employment of additional features, such as shape knowledge, texture, motion vectors, etc. As implemented here, we assume a-priori knowledge of the colours of the object to be isolated. Given a **N**-channel image $\mathbf{I}(\mathbf{I_1},\cdots,\mathbf{I_N})$, and a set of different colours/intensities $c=(c_1,c_2,\cdots,c_M)$. Then, $c_i,(i=1,\cdots,M)$ are vectors of length **N**. The components of the foreground and background colours of the k^{th} channel are $c^k_{fg}=(c^f_{k1},\cdots,c^f_{kR_f})$ and $c^k_{bg}=(c^b_{k1},\cdots,c^b_{kR_b})$, $\mathbf{R_f}+\mathbf{R_b}=\mathbf{M}$. Figure 1 gives an illustration. We choose an energy formulation with the following form:

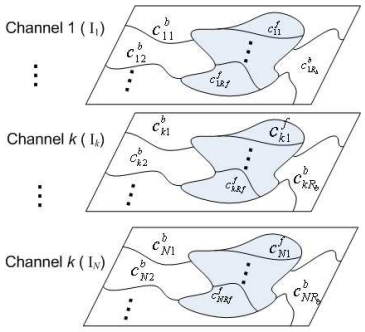


Figure 1: An image with N channels and a set of M different colours.

$$E(C)=\mu\cdot length(C)+\lambda_{\text{fg}}\cdot\iint_{\Omega_{\text{fg}}}F_{fg}(I(x,y),c_{\text{fg}})dxdy+\lambda_{\text{bg}}\cdot\iint_{\Omega_{\text{bg}}}F_{bg}(I(x,y),c_{\text{bg}})dxdy \quad (1)$$

where C is the boundary curve of Ω_{fg} (shaded in Fig.1). $\Omega_{fg} = c^f_{k1} \cup \cdots \cup c^f_{kR_f}$ is the foreground (object) which is inside C, and the complement of $\Omega_{bg} = c^b_{k1} \cup \cdots \cup c^b_{kR_b}$ is the background which is outside C. Then, according to the bin-by-bin dissimilarity measurement - Minkowski distance [9], we use the mean of L_2 (the standard deviation) and L_3 (the third root of the skewness) in each channel to get the expressions:

$$F_{fg}(I(x,y),c_{fg}) = \sum_{r=2}^{3}(\prod_{q=1}^{R_f}(\frac{1}{N}\sum_{p=1}^{N}(w^f_q|I_p(x,y)-c^f_{pq}|^r))^{1/r})^{1/R} \tag{2}$$

$$F_{bg}(I(x,y),c_{bg}) = \sum_{r=2}^{3}(\prod_{q=1}^{R_b}(\frac{1}{N}\sum_{p=1}^{N}(w^b_q|I_p(x,y)-c^b_{pq}|^r))^{1/r})^{1/R} \tag{3}$$

where $c_i = average(I_p(x,y))$ inside the i^{th} region. μ, λ_{fg} , λ_{bg} and $w^{f,b}_i (i = 1,\cdots,N)$ are nonnegative weights for the regularizing term and the fitting term, respectively. This model is robust to symmetric and asymmetric noise (e.g. Gaussian and Gamma distributed noise). The optimal partition is obtained by minimizing the energy $E(C)$. "*The key idea is to evolve the boundary C to the boundary of the object from some initialization in direction of the negative energy gradient under the constraints from the image.*"[7]

2.2 Level Set Formulation of the Model

For the level set formulation of the variational active contour model, we replace the unknown variable C by the unknown variable ϕ , and follow [10], using the Heaviside function H, and the one-dimensional Dirac measure δ_0 defined respectively by

$$H(z) = \begin{cases} 1 & , \quad if \quad z \geq 0 \\ 0 & , \quad if \quad z < 0 \end{cases} \qquad \delta_0 = \frac{d}{dz}H(z) \tag{4}$$

We express the terms in the energy E in the following way:

$$E(C) = \iint_{\Omega}(\mu \cdot \delta_0(\phi(x,y))|\nabla\phi(x,y)| + \lambda_{fg} \cdot F_{fg}H(\phi(x,y)) + \lambda_{bg} \cdot F_{bg}(1-H(\phi(x,y))))dxdy \tag{5}$$

In order to compute the associated Euler-Lagrange equation for the unknown function ϕ , our numerical simulations involve slightly regularized version of H and δ_0 , denoted here by H_ε and δ_ε , as $\varepsilon \longrightarrow 0$. In this paper, we approximate the regularization of Heaviside by the complementary error function (erfc).

$$H_\varepsilon(z) = \frac{1}{2}erfc\left(-\frac{\sqrt{\pi}z}{\varepsilon}\right) \qquad \delta_\varepsilon(z) = H'_\varepsilon = \frac{e^{-\left(\frac{\sqrt{\pi}z}{\varepsilon}\right)^2}}{\varepsilon} \tag{6}$$

This is very similar to the procedure used by [1][2] and [10], but it has a bigger support interval, $(-\infty,+\infty)$. Minimizing $E(C)$ with respect to ϕ yields the following Euler-Lagrange equation for ϕ , parameterizing the descent direction by time, $t > 0$. The equation in $\phi(t,x,y)$ (with $\phi(0,x,y) = \phi_0(x,y)$ defining the initial contour) is:

$$\frac{\partial\phi}{\partial t} = \delta_\varepsilon(\phi)\left[\mu \cdot \nabla \bullet \left(\frac{\nabla\phi}{|\nabla\phi|}\right) - \lambda_{fg}F_{fg} + \lambda_{bg}F_{bg}\right] \tag{7}$$

in Ω , and with the boundary condition $\frac{\delta_\varepsilon(\phi)}{|\nabla\phi|}\frac{\partial\phi}{\partial\vec{n}} = 0$ on Ω , where $\vec{n}$ denotes the normal at the boundary of Ω . Actually, $\frac{\nabla\phi}{|\nabla\phi|}$ is the unit (outward) normal, and the divergence of the normal $\nabla \bullet \left(\frac{\nabla\phi}{|\nabla\phi|}\right)$ is the mean curvature of the ϕ.

2.3 Numerical Implementation

To solve this evolution problem, we use the level set method proposed by Osher [8]. We define an implicit function for ϕ using a signed distance. This function is positive on the exterior, negative on the interior, and zero on the boundary. Meanwhile, an extra condition of $|\phi| = 1$ should be satisfied. ϕ does not have to be a signed distance function; for example a Euclidean distance transform or Chamfer distance transform could be chosen as a level set function ϕ. However, a signed distance function will increase the stability and quality of the evolution (especially if a vector field-based force and a force in normal direction are combined). This is because the signed distance is the path of steepest descent for the function. In order to improve numerical efficiency, we use a discrete form of the Hamilton-Jacobi (HJ) equation with high order ENO (Essentially Nonoscillatory) and WENO (Weighted ENO) accuracy and a Local Lax-Friedrichs (LLF) scheme. We also calculate the upwind derivative by using second order ENO scheme.

When working with level sets and Dirac delta functions, ϕ will no longer be a distance function (i.e. $|\phi| = 1$). ϕ can become irregular after some period of time. A standard procedure is to reinitialize the signed distance function to its zero-level curve. This prevents the level set function from becoming too flat, and can be seen as a rescaling and regularization. The reinitialization procedure is made by the following evolution equation:

$$\begin{cases} \psi_t = sign(\phi(t))(1 - |\nabla\psi|) \\ \psi(0,\bullet) = \phi(t,\bullet) \end{cases} \tag{8}$$

where $\phi(t,\bullet)$ is the solution ϕ at time t. Then the new $\phi(t,\bullet)$ will be ψ , such that ψ is obtained at the steady state of (8). The solution of (8) will have the same zero-level set as $\phi(t,\bullet)$ and away from this set, $|\nabla\psi|$ will converge to 1 [2].

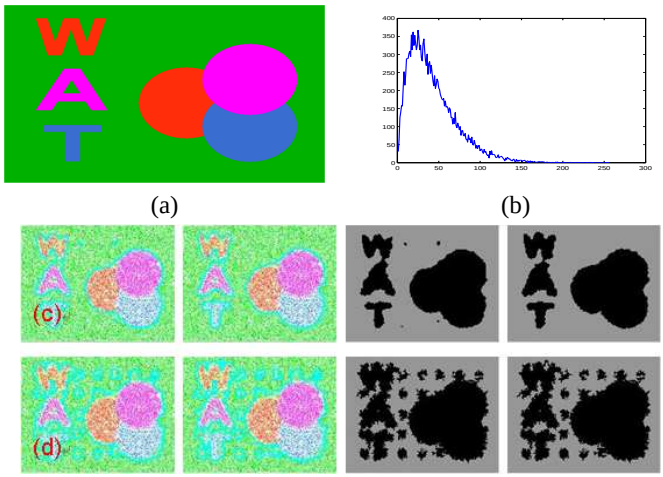


Figure 2: (a). Original synthetic image. (b). Gamma distributed noise. (c) and (d) show the results of iteration 40 and the final results by AS and CVV methods respectively; the noisy image is on the left, the associated piecewise-constant approximation on the right.

In all our numerical experiments, we generally choose the parameters: $\lambda_{fg} = \lambda_{bg} = 1$, $w_q^f = w_q^b = 1$. We use the approximations H_ε and δ_ε of the Heaviside and Dirac delta functions ($\varepsilon = \Delta x = \Delta y$), in order to automatically detect interior contours, and to insure the computation of a global minimizer. Only the length parameter μ , which has a scaling role, is not the same in all experiments. If we have to detect all or as many objects as possible and of any size, then μ should be smaller. Otherwise, μ should be larger. To test the effect of the L_3 Minkowski distance in the energy function, we first add asymmetric noise (i.e. a Gamma distribution) to a synthetic image. Fig.2(a) shows the original synthetic image, (b) shows the noise distribution, (c) shows the results of the AS method, and (d) the CVV method. This shows the improvement of the AS over the CVV method if the noise is additive and asymmetric. In Fig.3, each method is applied to a real image with a coloured, striped texture. This shows that the AS can obtain the complete contour, but the CVV has breaks in the expected segmentation. AS only needs 87 iterations to converge to the optimal solution, but the CVV method takes 202 iterations.

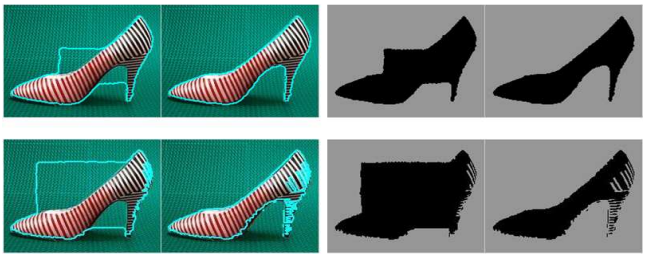


Figure 3: The comparison of the AS and CVV methods using a real image. The top figures are the results of iterations 60 and 87 by the AS method. The bottom figures are the results of iterations 60 and 202 by the CVV method. The original image is on the left and the associated piecewise-constant approximation is on the right.

We can also compare the accuracy of the AS method with that of the CVV method by calculating the energy of every evolution. Though energy formulation of the AS is different to that of CVV, and the initial value is different, we can compare the energy after normalization, because they should converge to the same global minimization, that is, $inf(E(C))$. For a perfect image and contours, $inf(F_{fg}) = inf(F_{bg}) = 0$, so $inf(E(C)) = \mu \cdot length(C)$. The comparison is shown in Figure 4. AS/CVV (Three colours) means we consider the three overlapping circles in Figure 2(a) as a single object and use the AS/CVV method. AS/CVV (one or two colours) has similar meaning. The experimental results show that the AS method only needs a small number of iterations to reach the minimum energy value. For example, for the object with three colours, the initial energy of the AS is bigger than that of CVV. After 25 iterations,

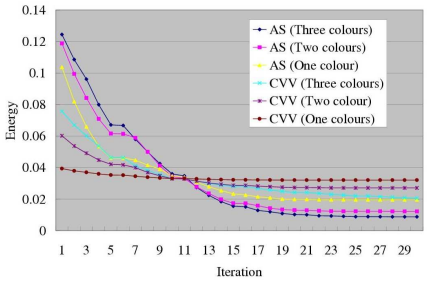


Figure 4: Comparisons of the energy evolution.

AS obtains a minimum, but the CVV method requires 150 iterations to obtain its final minimum.

3 Adaptive Object Tracking

3.1 Outline of the Adaptive Tracking Algorithm

The adaptive tracking algorithm is expressed as pseudocode,

- **Define the internal and external rectangles covering the object centroid at y0 in the first image.**
- **Sort *CSM*s by similarity distance criterion (Eq.10).**
- **Choose preferred *CSM*.**
- **Get active contour and ϕ of the tracked object by AS method.**
- **Repeat**
  - **Input the image i (initial value $i = 1$).**
  - **Obtain the set of foreground and background pixels by ϕ .**
  - **Sort and choose preferred *CSM*.**
  - **Get the sets of constant colours by clustering using mean-shift segmentation.**
  - **Compute NCDT kernel using Chamfer distance transform.**
  - **Form model histogram, q, in the preferred colour space.**
  - **Fetch the next frame $i+1$.**
    - **Compute candidate histogram $p(y_0)$ in the preferred *CSM*s using NCDT-kernel**
    - **Find the optimum location y_1 of candidate using mean shift tracking algorithm.**
    - **Get the motion vector.**
  - **Translate the contours.**
  - **Update ϕ by AS method.**
  - $i = i+1$.
- **Until end of input sequence**

3.2 Selection of the Best Colour Space Model

In tracking an object through a colour image sequence, we shall assume that we can represent it by use of a discrete distribution of samples from a region in colour space, initially localised by a kernel whose centre defines the current position. Hence, we want to find the maximum in the distribution of a function, ρ, that measures the similarity between the weighted colour distributions as a function of position (shift) in the *candidate* image with respect to a previous model image. If we have two sets of parameters for the respective densities $p(x)$ and $q(x)$, the Bhattacharyya coefficient is an approximate measurement of the amount of overlap, defined by [6]:

$$\rho(y) = \rho[p(y), q] = \sum_{u=1}^{m} \sqrt{p_u(y) q_u} \tag{9}$$

The distance between two distributions can be defined as

$$d(y) = \sqrt{1 - \rho(y)} \tag{10}$$

Clearly the distance $d(y)$ lies between zero and unity, and obeys the triangle inequality. In a discrete space, $x_i, i = 1, 2, \cdots, n$ are the pixel locations of the model, centred at a spatial location $\mathbf{0}$, which is defined as the position of the window in the preceding frame

that we want to track. A function $b : R^2 \rightarrow 1,2,\cdots,n$ associates to the pixel at location x_i the index $b(x_i)$ of the histogram bin corresponding to the value of that pixel. Hence a normalized histogram of the region of interest can be formed (using q_u as an example)

$$q_u = \frac{1}{n}\sum_{i=1}^{n}\delta[b(x_i)-u], \quad u = 1,2,\cdots,m \tag{11}$$

where δ is the Kronecker delta function. Tracking success or failure depends primarily on how distinguishable an object is from its surroundings. If the object is very distinctive, it is easy to track. Otherwise it is hard to track. Normally, the features that best distinguish between foreground and background are the best features for tracking. The choice of feature space will need to be continuously re-evaluated over time to adapt to changing appearances of the tracked object and its background. To select the best colour space model (*CSM*), we sort the different *CSM*s using the Bhattacharyya coefficient which is an approximate measurement of the amount of overlap between the two distributions of foreground and background. For the first frame, we use a "centre-surround" approach to sample pixels from object and background. A rectangular set of pixels covering the object is chosen to represent the object pixels, while a larger surrounding ring of pixels of the rectangle is chosen to represent the background. For an internal rectangle of size $h \times w$ pixels, the outer margin of width $(\sqrt{2}-1)\sqrt{hw}/2$ pixels forms the background sample. The foreground and background have the same number of pixels if $h = w$. In all subsequent frames, it is the contour defined by the level set function, ϕ, that defines the foreground for the adaptive model, so that no background is included. We use the distance criterion (10) to measure the similarity between the two histograms of the internal and external regions. The best colour space is selected by finding the *CSM* with maximum distance value. Each potential feature set typically has dozens of tunable parameters and therefore the full number of potential features that could be used for tracking is enormous. We construct 16 single frame *CSM*s from 5 different colour spaces (R.G.B, L.a.b, H.S.V, Y.I.Q/Y.Cb.Cr, C.M.Y.K). All the values of pixels are normalized to 0 to 255, yielding feature histograms with 16 or 256 bins.

Fig.5(a) shows a sample image with concentric boxes delineating the object and background. The similarity distances between foreground and background of each *CSM* are shown in Fig.5(b) and the set of all 16 candidate images after rank-ordering the feature according to the criterion (10) are shown in Fig. 5(c). The image with the most discriminative feature (best for tracking) is at the upper left. The image with the least discriminative feature (worst for tracking) is at the lower right.

3.3 Using a Kernel Based on the Normalized Chamfer Distance Transform

A radially symmetric kernel K can be described by a 1D profile rather than a 2D (or higher order) image. The most popular choice for K is the optimal Epanechnikov kernel that has a uniform derivative of $G = 1$ which is also computationally simple. However, in tracking an object through a video sequence and applying the mean shift algorithm to move the position of the target window, the bounds of the domain R^2 are altered on each successive application of the algorithm. There is no reason to suppose that the target has radial symmetry, and even if an elliptical kernel is used, i.e. there is variable bandwidth,

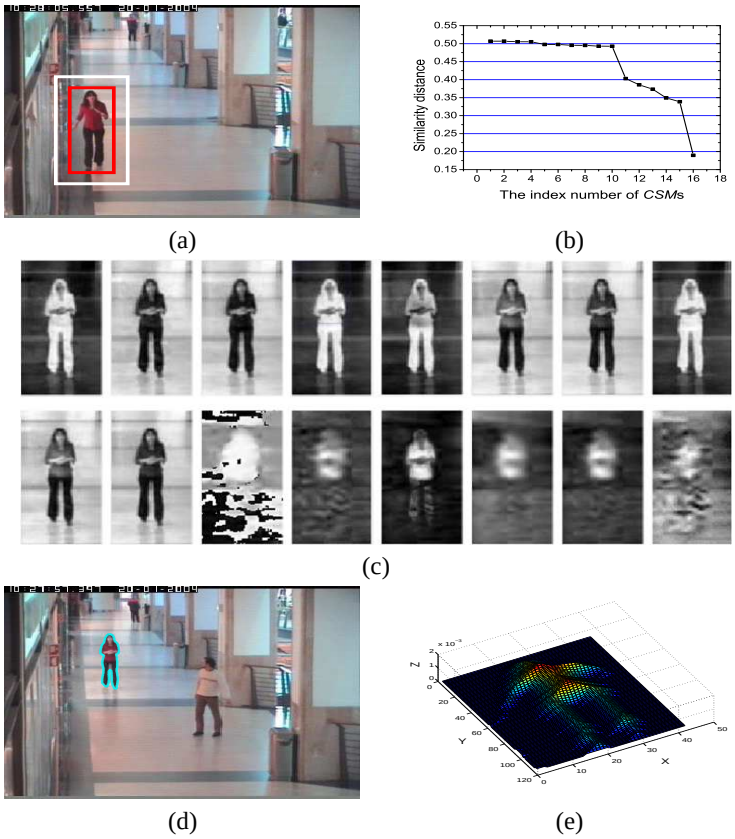


Figure 5: (a). A sample image with concentric boxes delineating the object and background. (b). The similarity distance of each *CSM*. (c). Rank-ordered 16 images. (d). AS segmentation result. (e). 3D NCDT kernel

the background area that is being sampled for the colour distribution will change. If the background is uniform this will not affect the colour *pdf*, and hence the gradient ascent will be exact. However, if it is not uniform, but varies markedly and in a worst case has similar properties to the target, as we shall see in the next section, then multiple modes will be formed in the *pdf* and the mean shift is no longer exact. Therefore, we use the normalised Chamfer distance transform (NCDT) rather than the true Euclidean distance, as it is an efficient approximation. The NCDT kernel better represents the colour distribution of the tracked target, yet retains the more reliable centre weighting of the radially symmetric kernels. This transform is applied to the target area, separated from the background by AS methods described in Section 2.3. Figures 5(d) and (e) show the AS segmentation and the NCDT kernel of Figure 5(a). We aim to show that this weighting increases the accuracy and robustness of representation of the *pdf*'s as the target moves, since it excludes peripheral pixels that occur within a radially symmetric window applied to successive frames. We are investigating the performance of the NCDT

transform to define the regions of interest and weight the colour densities in the video images. We assess whether the anticipated gain in excluding background pixels from the density estimates and weighting more substantially those more reliable pixels towards the centre of the tracked object will outweigh the possibility of forming false modes because of the shape of the NCDT. However, radially symmetric kernels may also produce false modes due to badly defined densities.

Figure 6 illustrates that the tracking algorithm can cope with dynamic deformation of the shapes and the changing positions of the targets in the various sequences, even when the camera pans so that both the foreground and background move in the camera coordinate system (Fig. 6(a)). All of these illustrations are from much longer sequences, included as supplementary material, typically more than a hundred frames. Fig.6(b) and (c) show that the tracking algorithm is very robust to clutter, and crossing objects. The car is occluded by a square object which has the same colour as the car in the third sample picture in (b), two people cross in the third sample picture in (c), yet the algorithm adapts the contour to track the non-occluded portion of the woman, then re-grows the contour as she re-emerges from behind the man. In each of the sequences, the rectangle in the first image defines the initial region, in which the object to be tracked is segmented. In Figure 6(c), we have also compared the use of the NCDT and Epanechnikov kernel, but in the latter case the tracker latches on the crossing individual.

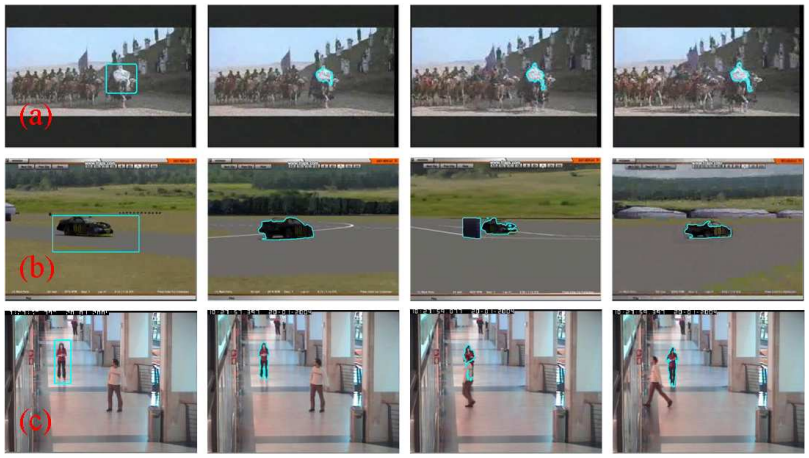


Figure 6: Tracking video objects and dynamics of deformation. The video sequences are supplied as supplementary material.

4 Conclusions

We have developed segmentation and tracking algorithms using a generalized active contour model. The object of interest can have a mixture of colours, but these are known before segmentation. For segmentation, the position of the initial curve can be anywhere in the image, and need not necessarily surround the object to be detected. However, if the

initial estimate is far from the true contour, it takes a long time to converge to the optimal solution. The segmentation is similar to the earlier CVV algorithm, but uses a slightly different cost function that deals better with noise that is asymmetric, and converges more quickly on sample image data. The adaptive object-tracking is a hybrid algorithm, combining level set methods with the mean shift tracking algorithm. Mean shift defines the translation in the next frame to accelerate the level set definition of the tracked contour. The algorithm is also improved by a Chamfer distance transform (NCDT kernel) and sorted CSMs to better detect and track objects. Several experiments have demonstrated the ability of the model to detect and track an object in movie sequences.

References

[1] T. Chan, B.Y. Sandberg, and L. Vese. Active contours without edges for vector-valued images. *Journal of Visual Communication and Image Representation*, 11(2):130–141, 2000.

[2] T.F. Chan and L.A. Vese. Active contours without edges. *IEEE Trans. on Image Processing*, 10(2):266–277, 2001.

[3] J.S. Chang, E.Y. Kim, K. Jung, and H.j. Kim. Object tracking using mean shift and active contours. In *Proceedings of the 18th Int. Conf. on Innovations in Applied Artificial Intelligence*, pages 26–35, Bari, June 2005.

[4] R.T. Collins, Y. Liu, and M. Leordeanu. Online selection of discriminative tracking features. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(10):1631–1643, 2005.

[5] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[6] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(5):564–577, 2003.

[7] D. Cremers, M. Rousson, and R. Deriche. A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape. *International Journal of Computer Vision*, 72(2):195–215, 2007.

[8] R. Osher, S.and Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 2003.

[9] Y. Rubner, C. Tomasi, and L.J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[10] H.K. Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *Journal of Computational Physics*, 127(1):179–195, 1996.

Non-Gibbsian Markov random field models for contextual labelling of structured scenes

Daniel Heesch and Maria Petrou
Imperial College London
Communications and Signal Processing Group
Department of Electrical and Electronic Engineering
London SW7 2AZ, UK
{daniel.heesch,maria.petrou}@imperial.ac.uk

Abstract

In this paper we propose a non-Gibbsian Markov random field to model the spatial and topological relationships between objects in structured scenes. The field is formulated in terms of conditional probabilities learned from a set of training images. A locally consistent labelling of new scenes is achieved by relaxing the Markov random field directly using these conditional probabilities. We evaluate our model on a varied collection of several hundred hand-segmented images of buildings.

1 Introduction

Recent years have seen notable improvements in the performance of object classifiers. Greater robustness against occlusion and intraclass variability has been achieved by describing objects by a large number of local and largely view-invariant features (e.g. [15, 5, 18, 14]). For single classes efficient classification methods such as boosting allow recognition to be in real-time (e.g. [17]). Some of these models have the additional benefit of biological plausibility. The hierarchical feed-forward architecture of [13] aims to mimic the ventral stream of visual information processing and is able to predict with great accuracy whether or not an object is present in a scene.

It seems, however, that in order to be able to scale to the several thousands of categories humans discriminate without effort, appearance based object classification needs to be complemented by techniques that utilise contextual information. Context may be described as any dependency between the object to be recognised and everything else in the scene, be these other objects or the scene as a whole. Experimental evidence suggests that humans do exploit both types of dependency during object recognition. It is well established, for example, that the nature of a scene can be recognised based on low spatial frequency information [11]. Recent neuro-imaging studies support the view that low spatial frequencies are processed in the cortex at a very early stage during visual recognition [2], suggesting that perception involves top-down facilitation. Much like the gist of a scene, the spatial relationships between objects can be determined without high frequency information. Bar and Aminoff in [1] establish early activation of cortical "context networks" that appear to store spatial relationships, pointing to a key role of spatial context as an early facilitator during object recognition.

Our goal is to learn these spatial and topological relationships from the data and to utilise this information in a Markov random field (MRF) model to achieve a consistent labelling of new scenes. The MRF is defined not over a pixel array but the set of regions that correspond to objects. From training data we learn the probability distribution over labels for a region, given the objects in its local neighbourhood. These supply the conditional probabilities that define the MRF and are used during an iterative relaxation scheme to find a probable realisation given the structural relationships observed in a new scene.

Unlike the MRFs hitherto used in computer vision, the MRFs we use here are non-Gibbsian, i.e. they cannot be expressed in terms of cliques and a global cost function. This is because the interactions between units are directional and non-symmetric (A influences B differently from how B influences A). Such MRFs are characteristic of natural complex systems and they may be used to model, for example, the interaction between neurons in the human brain, population dynamics or company interactions. Complex systems subject to such unit interactions tend to oscillate between different states rather than converge to a single state [9]. In the case of human perception, the human brain is then somehow able to select from the possible interpretations the most appropriate one. In this paper we use a relaxation method appropriate for producing the states of such an MRF and a criterion that allows us to select the right state.

We validate our approach on a set of about 250 photographs of buildings that were manually segmented and labelled. This domain is particularly interesting as it exhibits sufficiently tight structural constraints to benefit from our approach, and a fair amount of structural variability to challenge it.

This paper is structured as follows. Section 2 presents related work. Section 3 introduces the non-Gibbsian model. Section 4 details how it is used to label new scenes. Section 5 describes a series of experiments to validate our approach. Section 6 concludes the paper.

2 Related work

We here consider related works that are concerned with modelling peer-to-peer, rather than hierarchical, dependencies. A natural choice for probabilistic modelling of local dependencies are Markov random fields [8], defined either on a segmentation of the image as in [10, 4] or on a rectangular grid as in [7, 6, 14]. The authors in [6] and [14] define a conditional random field over individual pixels. In [14], contextual information is incorporated by using the joint boosting algorithm [16] for learning potential functions and by employing a novel feature that captures local dependencies in appearance. Neither work explicitly considers spatial relationships, although in [6] the absolute position of a site is included in the potential function.

In [4], it is assumed that training images are associated with a bag of words with no explicit mapping between regions and terms. This renders the learning task more difficult but makes it easier to get hold of large amounts of training data. The MRF is specified through single and pair-wise clique potential functions learned from the data. To make the estimation problem tractable, potential functions are symmetric with respect to their arguments (labels of adjacent image regions). The model does not capture asymmetric dependencies, nor does it take into account spatial relationships.

In [10], an MRF is defined over image regions by specifying the clique functions for all types of single and pair-wise cliques. The potential functions are taken to be a weighted sum of m basis functions whose parameters are set manually.

Our objectives are similar to those in [4] and [10]. Unlike those two, however, we allow neighbouring blobs to influence each other differently depending on their relative spatial position. The asymmetry thereby introduced forbids the definition of cliques and thus the formulation of the MRF in terms of a Gibbs distribution. Our model consists of conditional probabilities that are learned directly from the data using structural information as can be obtained from the low spatial frequency content of an image.

3 The model

3.1 Non-Gibbsian MRF

Let $S = \{1,\ldots,N\}$ index a set of regions in an image. We assume that each region is associated with a random variable f_i which takes its value from a discrete set of class labels. The field $F = \{f_i : i \in S\}$ is assumed to be Markovian in the sense that the probabilistic dependencies among f_i are restricted to spatial neighbourhoods $\mathcal{N}_i$, that is,

$$P(f_i|f_{S-i},R) = P(f_i|f_{\mathcal{N}_i},R_i), \tag{1}$$

where R denotes the matrix of pair-wise spatial relationships between regions, and R_i the row pertaining to region i. We assume, therefore, that the conditional dependencies depend not only on the identity of the neighbouring regions but also on their relative spatial relationships with the ith region. This is an important component of our model as it allows us to capture the non-isotropic nature of many scenes. For convenience, we refer to a particular observation pair $(f_{\mathcal{N}_i},R_i)$ as the *neighbourhood configuration* or simply *configuration*, and to the ith region associated with it as the *focal region*.

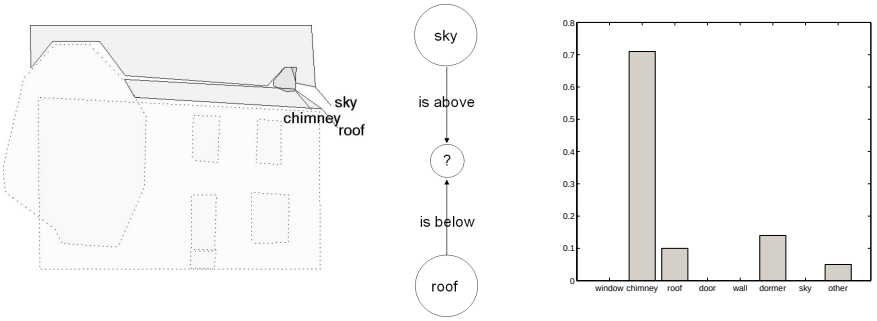


Figure 1: A particular configuration associated with a chimney (left), a schematic representation of the configuration $(f_{\mathcal{N}_i},R_i)$ (middle) and the conditional probability distribution over all labels associated with that configuration, $P(f_i|f_{\mathcal{N}_i},R_i)$, as obtained from training images (right). The distribution tells us that a region below sky and above a roof is a chimney (71%) but may also be a dormer (14%) or another roof (10%).

3.2 Neighbourhoods

Since we need to learn the conditional distributions from a relatively small training set, we limit the neighbourhood to at most six regions: the neighbour above, below, to the left and to the right of region i, as well as the region containing and being contained by region i. The

neighbourhood relation is reciprocal and two regions are neighbours if they are separated by no more than a certain distance threshold. The distance between two regions $A, B \subset \mathbb{R}^2$ is computed as

$$d(A,B) = \sum_{i \in \{x,y\}} \min_{a \in A, b \in B} |a_i - b_i|, \tag{2}$$

where a_x represents the x coordinate of point a. Other choices of a distance function are of course conceivable. This particular one has the effect that two regions need not be the same to have a zero distance but may be (i) overlapping, (ii) exactly adjacent or (iii) contained in one another. For example, a wall that surrounds a number of windows has a zero distance from each of them. If regions are non-overlapping, the distance along each direction is given by the smallest Euclidean distance between any two points of the two regions. This has the advantage that the distance between two regions is not affected by their respective sizes (as would be the case under many metrics such as the Hausdorff metric). For a distance cutoff of 0, the neighbourhood consists of all regions whose bounding boxes overlap with or touch the focal region. Were the regions regularly arranged like pixels, the resulting neighbourhood would be the familiar 8-pixel neighbourhood. The optimal distance cutoff is learned through cross-validation. Figure 2 depicts the distribution over configuration sizes for the optimal zero cutoff. The right figure illustrates how the configurations become larger as the distance cutoff increases.

Given a distance threshold, the conditional probability distributions (eq. 1) are learned by noting for each region i observed in a set of training images its corresponding configuration $(f_{\mathcal{N}_i}, R_i)$. The results can conveniently be stored in the form of a hashtable with the key being a particular configuration and the value being the conditional probabilities over labels for the focal region. Given a region with known neighbourhood configuration, we can thus rapidly obtain a probability distribution over labels at the focal region. To ensure that the joint distribution of the MRF is nowhere zero, we add a small positive value to each zero-valued conditional probability and subsequently normalise.

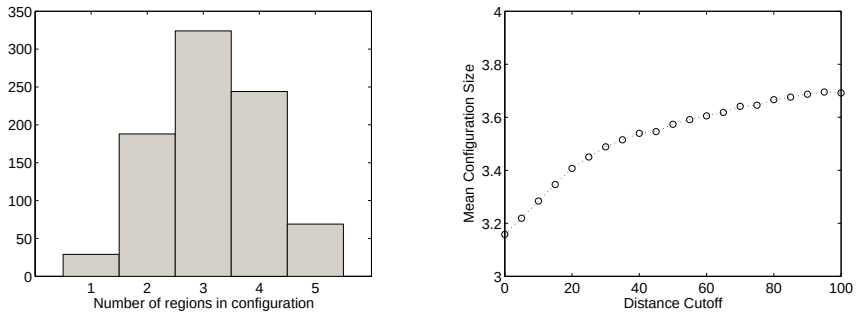


Figure 2: Frequency distribution of different configuration sizes for a distance cutoff of zero (left). As we increase the distance threshold, the configurations become larger (right).

4 Labelling of new scenes

This section details how to obtain probable realisations of the MRF given a new scene. We make the assumption that scenes have been segmented into regions where each region corresponds to an object to be recognised. How these regions are obtained in the first place

is a problem in its own right and outside the focus of this work. We shall simply take it for granted that an appropriate segmentation has been achieved.

4.1 Global Gibbsian versus local non-Gibbsian relaxation

A standard technique to find a probable realisation of an MRF is simulated annealing which allows a stochastic label update at a site to be retained with a certain probability P_r even if the new realisation of the field is less probable. By letting P_r converge to zero, the field eventually settles at a maximum of the joint probability distribution. In other words, simulated annealing strives to find solutions that are globally maximally consistent.

Because of the impossibility to define cliques, our non-Gibbsian field is formulated purely in terms of local, conditional probability distributions (Equation 1). We aim to find labellings that are locally consistent by repeatedly sampling from these conditional distributions.

4.2 Graph colouring

In order to iteratively update regions based on the current labelling of their neighbourhood, we partition the set of regions into a set of codings. The idea of a coding was first introduced by Besag [3] in the context of the iterated conditional mode algorithm for MRF parameter estimation. A coding is equivalent to the concept of a vertex colouring of a graph, that is, it constitutes a partitioning of the set of vertices (= regions) so that no two adjacent vertices (= neighbouring regions) belong to the same partition. Because of the assumption of Markovianity, the likelihood over vertices of the same colour reduces to a simple product of the respective conditional probabilities. We employ a greedy strategy to achieve a vertex colouring, in which vertices are visited in order of decreasing vertex degree (i.e. number of neighbours). Each vertex is assigned the first possible colour from a list of colours. One example of a colouring is given in Figure 3. The wall has the largest number of neighbours and is correspondingly assigned the first colour (‘1’).

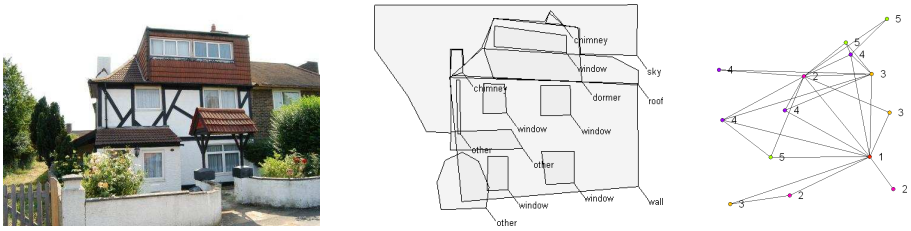


Figure 3: Original image (left). Hand-segmented and hand-labelled training image (middle). Vertex colouring of the neighbourhood graph (right): vertices with the same number have non-overlapping neighbourhoods.

4.3 Choosing a solution

Regions are updated within each coding by retrieving and sampling from the probability distribution corresponding to that region’s current neighbourhood configuration. If the configuration has not been seen before, because it was not observed in the training set, the new label is drawn from a uniform distribution. This scheme on its own is not guaranteed to

converge and indeed it seems to have no tendency to do so. Following each update, we compute for each coding $\mathscr{C}_j$

$$P(f_{\mathscr{C}_j}|R) = \prod_{i \in \mathscr{C}_j} P(f_i|f_{\mathscr{N}_i}, R)$$

Our estimate of the overall probability of the data is obtained by averaging over $P(f_{\mathscr{C}_j}|R)$. Because the codings are generally of different size, the arithmetic average sometimes used for regular MRF is unsuitable. Instead, we estimate the joint probability as

$$P(f_1, \ldots, f_N) \approx \frac{1}{N} \sum_j |\mathscr{C}_j| \left[\prod_{i \in \mathscr{C}_j} P(f_i|f_{\mathscr{N}_i}, R) \right]^{\frac{1}{|\mathscr{C}_j|}}. \quad (3)$$

Let p be the ratio between the estimated joint probability after and before the update. We accept the change with probability 1 if $p > 1$ and with probability $p^{\frac{1}{T}}$ otherwise. T is the temperature parameter whose value decreases exponentially with time. Figure 4 shows an example of how the value given by eq. 3 increases over successive iterations. One iteration here involves the update of the labels of all regions.

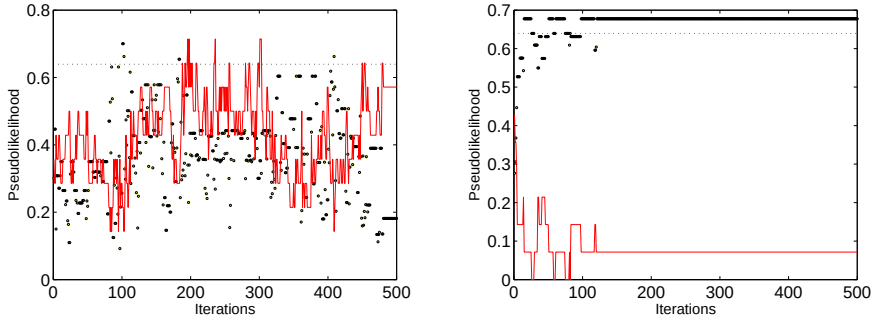


Figure 4: Dynamics of stochastic updating process with and without maximisation of the pseudolikelihood. The dotted line marks the pseudolikelihood associated with the true labelling. The continuous line shows the proportion of misclassified regions. In both diagrams, regions are updated based on the conditional probabilities. For the left diagram, a new labelling is always accepted, for the right diagram, a labelling is accepted when it improves the current optimum or when it is worse by no more than a value that decreases with time.

5 Experiments

For our experiments, we collected 253 images of buildings from the World Wide Web. Each image was manually segmented into regions that correspond to parts of the building or parts of the environment such as sky or vegetation. Each region was labelled by hand using an annotation tool similar to LabelMe. The complete dataset contains nearly 6,000 regions covering a dozen of classes.[1]

[1]The images along with the annotation and segmentation information is available at http://www.commsp.ee.ic.ac.uk/∼dheesch/ngmrf/data/

We allow for the following seven labels (with respective frequencies): ‘window’ (0.507), ‘chimney’ (0.054),‘roof’ (0.053), ‘door’ (0.087), ‘wall’ (0.089) , ‘dormer’ (0.015), ‘sky’ (0.055), ‘other’ (0.14). The ‘other’ label aggregates all remaining structures that were annotated (e.g. ’pipes’ and ’balcony’). We report performance of different algorithms in terms of classification accuracy, i.e. the proportion of regions that have been labelled correctly. To estimate how the algorithm will be able to predict data that it was not trained on, we use the leave-one-out method of cross-validation, i.e. we remove one image from the set at a time to be our test image and train on the remaining 252 images.

5.1 Comparison with other methods

We compare our non-Gibbsian MRF model with two other classification models, a non-contextual Bayes classifier and an alternative contextual model that uses probabilistic relaxation to find a locally consistent labelling.

5.1.1 Non-contextual Bayes classifier

As a non-contextual benchmark we implemented a Parzen classifier that classifies regions based on the posterior probabilities given measurements of a number of low-level features from the region. We use a set of three features that can easily be obtained from the low-frequency content of a scene: the mean intensity, the normalised area of the region and its vertical position. For each feature, the posterior probabilities over classes is given by Bayes rule with the class-conditional densities being approximated using a Parzen window with a Gaussian kernel function centred on a set of class exemplars E_c

$$p(x|c) \propto \sum_{x_i : i \in E_c} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{|x-x_i|}{2\sigma^2}\right), \tag{4}$$

where σ is learned through cross-validation. We assume each feature to be conditionally independent given the class, and thus compute the overall class probability density as a product of feature-specific posteriors.

5.1.2 Probabilistic relaxation

The second comparison is with an alternative contextual labelling technique known as probabilistic relaxation [12]. The contextual information consists of the conditional probabilities of a label, given that another label is found in a particular relative position to the first. In each iteration of the relaxation process, the label probabilities are updated based on the probabilities at the previous time step, modulated by the support a particular label f_i receives from neighbouring labels,

$$P^{(n+1)}(f_i = c) = \frac{P^{(n)}(f_i = c) Q_i(c)}{\sum_{\mu \in \mathscr{L}} P^{(n)}(f_i = \mu) Q_i(\mu)} \tag{5}$$

with support function

$$Q_i(c) = \sum_{j : f_j \in \mathscr{N}_i} \sum_{\nu \in \mathscr{L}} P(f_i = c | f_j = \nu, r_{ij}) P(f_j = \nu). \tag{6}$$

Here $\mathscr{L}$ denotes the label set. The compatibilities are learned from the data in a similar way as are the conditional distributions for neighbourhood configurations in the MRF model. Note that unlike the MRF model, which allows configurations to comprise up to six regions, this particular formulation of probabilistic relaxation is limited to binary dependencies. This makes statistical learning of dependencies easier but comes at the expense of limited modelling power.

5.1.3 Results

Table 1 shows the results for probabilistic relaxation and our NG-MRF when using the output of the Parzen classifier to initialise the labelling. In order to assess the variability in performance, we have opted for a leave-one-out strategy. The results are the average over 253 images with more than 5,000 regions.

The best results are obtained by the non-Gibbsian MRF, followed closely by the non-contextual classifier. It is noteworthy that this particular version of probabilistic relaxation, instead of improving the results of the non-contextual Parzen method, makes them worse.

Regions	Unique cfgs	Prior	Parzen	PR	NG-MRF
5,682	0.904	0.521 (0.0006)	0.690 (0.125)	0.568 (0.134)	0.729 (0.124)

Table 1: Performance comparison for different classification methods. Prior: each region is given the same, most frequently occuring label; Parzen: non-contextual classification; PR: probabilistic relaxation; NG-MRF: non-Gibbsian Markov random field. Performance is measured in terms of the proportion of regions classified correctly (standard deviation in brackets). The second column gives the proportion of unique configurations in the test set for which a conditional distribution has been learned from the training images.

Table 1 does not show how performance varies between different classes. As the confusion matrix in Table 2 indicates, by far the greatest accuracy is achieved for windows. That many other classes are misclassified as windows may be attributed to the strong prior on the 'window' class that influences the result through the non-contextual Parzen initialisation. Note that doors in particular are frequently mistaken for windows as these two classes exhibit very similar spatial relationships with other building parts whilst having markedly different priors.

	wi	ch	ro	do	wa	do	sk	ot
window	2848	50	5	81	0	0	25	131
chimney	20	151	50	5	0	5	10	15
roof	25	20	101	0	30	10	25	76
door	348	5	0	20	5	0	0	96
wall	40	0	25	5	292	10	10	91
dormer	30	15	20	5	5	15	5	0
sky	15	10	10	0	5	5	192	30
other	217	15	15	40	30	5	25	343

Table 2: Confusion matrix for NG-MRF labelling. The top row entries are indexed by the first two letters of the respective label. The matrix element a_{ij} gives the number of regions of the ith class that have been classified as belonging to the jth class.

5.2 Robustness to initialisation

We investigate two different initialisation schemes to assess the robustness of the contextual inference to initial conditions. The first scheme assigns each region the most frequently occuring label (in this case 'window'), the second draws labels randomly from the prior distribution, i.e. it will result in a similar initial distribution of classes within the image but with random assignment of classes to regions. The results are shown in Table 3. While we notice a performance degradation compared to non-contextual initialisation, the contextual model continues to improve over the new baselines of 0.52 and 0.32, respectively.

Initialisation scheme	Initial	NG-MRF
Non-contextual	0.690	0.729 (0.124)
Max Prior	0.521	0.654 (0.127)
Random	0.315	0.621 (0.135)

Table 3: Dependence of contextual classification on initial conditions. The second column shows the accuracy after initialisation with the three different schemes discussed in the text. The initial accuracy of the random assignment is $1-\sum_c p_c(1-p_c)$ where p_c is the prior of the cth class.

6 Conclusions

We presented a Markov random field model for contextual labelling of objects in structured scenes. In our model the context of a region consists not only of the identity of neighbouring regions but also, crucially, on their relative spatial and topological relationships. By incorporating what are typically asymmetric relationships, the Markov random field is capable of modelling the non-isotropic nature of typical scenes. The asymmetry makes the field non-Gibbsian as it no longer admits to a factorisation into cliques, so that the model is formulated in terms of conditional distributions that are learned from training data.

Given a new scene, the Markov random field is relaxed by iteratively sampling from conditional probability distributions. We proposed an objective function to help us identify good labelling solutions. The objective function is based on the vertex colouring of the region neighbourhood graph and is not the global cost function usually associated with Gibbsian MRFs. A comparison with a non-contextual and an alternative contextual classifier suggests the validity of the approach.

There are several ways how to take the work further. For this study we hand-segmented and hand-labelled several hundred images. To demonstrate the robustness of the technique, a next step is to learn configurations from automatically segmented, but possibly hand-labelled training exemplars. Also, we currently make no attempt to generalise from observed configurations to new ones. As some configurations are supersets of smaller configurations, or are otherwise similar to each other, endowing the configuration space with some distance metric would allow more accurate label distributions to be inferred for previously unseen configurations.

References

[1] M Bar and E Aminoff. Cortical analysis of visual context. *Neuron*, 38:347–358, 2003.

[2] M Bar, K Kassam, A Ghuman, J Boshyan, A Schmidt, A Dale, M Hämäläinen, K Marinkovic, D Schacter, B Rosen, and E Halgren. Top-down facilitation of visual recognition. *Proceedings National Academy of Sciences*, 103(2):449–454, 2006.

[3] J Besag. Spatial interaction and the statistical analysis of lattice systems (with discussion). *Journal Royal Statistical Society, B*, 36:192–236, 1974.

[4] P Carbonetto, N de Freitas, and K Barnard. A statistical model for general contextual object recognition. In *Proc European Conf Computer Vision*, pages 350–362, 2004.

[5] G Csurka, C Bray, C Dance, and L Fan. Visual categorization with bags of keypoints. In *Proc European Conf Computer Vision*, 2004.

[6] X He, R Zemel, and D Ray. Learning and incorporating top-down cues in image segmentation. In *Proc European Conf Computer Vision*, 2006.

[7] S Kumar and H Hebert. Discriminative random fields: a discriminative framework for contextual interaction in classification. In *Proc Int'l Conf Computer Vision*, 2003.

[8] S Li. *Markov Random Field Modeling in Computer Vision*. Springer, New York, 1995.

[9] Z Li and P Dayan. Computational differences between asymmetrical and symmetrical networks. *Network: Computation in Neural Systems*, 10(1):59–77, 1999.

[10] J Modestino and J Zhang. A Markov Random Field model-based approach to image interpretation. *IEEE Trans Pattern Analysis and Machine Intelligence*, 14(6):606–615, 1992.

[11] A Oliva and A Torralba. Modelling the shape of the scene: a holistic representation of the spatial envelope. *Int'l Journal Computer Vision*, 42(3):145–175, 2001.

[12] A Rosenfeld, A Hummel, and S Zucker. Scene labeling by relaxation operations. *IEEE Trans Systems, Man and Cybernetics*, 6(6):420–433, 1976.

[13] T Serre, A Oliva, and T Poggio. A feedforward architecture accounts for rapid categorization. *Proceedings National Academy of Science*, 104(15):6424–6429, 2007.

[14] 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 European Conf Computer Vision*, 2006.

[15] J Sivic and A Zisserman. Video google: a text retrieval approach to object matching in videos. In *Proc Int'l Conf Computer Vision*, pages 1–8, 2003.

[16] A Torralba, K Murphy, and W Freeman. Sharing fatures: efficient boosting procedures for multiclass object detection. In *Proc Int'l Conf Computer Vision and Pattern Recognition*, pages 762–769, 2004.

[17] P Viola and M Jones. Rapid object detection using a boosted cascade of simple features. In *Proc Int'l Conf Computer Vision and Pattern Recognition*, 2001.

[18] J Winn, A Criminisi, and T Minka. Object categorization by learned universal visual dictionary. In *Proc Int'l Conf Computer Vision*, pages 1800–1807, 2005.

Denoising Manifold and Non-Manifold Point Clouds

Ranjith Unnikrishnan Martial Hebert
Carnegie Mellon University, Pittsburgh, PA 15213
ranjith,hebert@cs.cmu.edu

Abstract

The faithful reconstruction of 3-D models from irregular and noisy point samples is a task central to many applications of computer vision and graphics. We present an approach to denoising that naturally handles intersections of manifolds, thus preserving high-frequency details without oversmoothing. This is accomplished through the use of a modified locally weighted regression algorithm that models a neighborhood of points as an implicit product of linear subspaces. By posing the problem as one of energy minimization subject to constraints on the coefficients of a higher order polynomial, we can also incorporate anisotropic error models appropriate for data acquired with a range sensor. We demonstrate the effectiveness of our approach through some preliminary results in denoising synthetic data in 2-D and 3-D domains.*

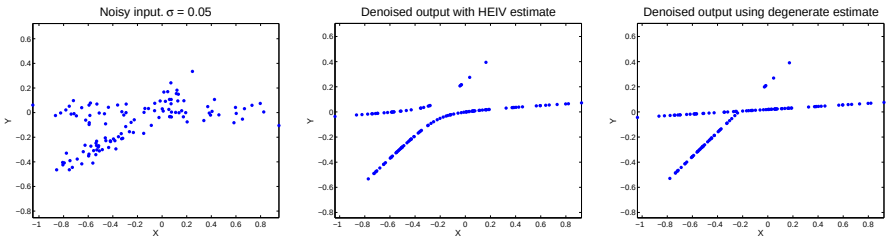
1 Introduction

Surface reconstruction from unorganized point samples is a challenging problem relevant to several applications, such as the digitization of architectural sites for creating virtual environments, reverse-engineering of CAD models from probed positions, remote sensing and geospatial analysis. Improvements in scanner technology have made it possible to acquire dense sets of points, and have fueled the need for algorithms that are robust to noise inherent in the sampling process.

In several domains, particularly those involving man-made objects, the underlying geometry consists of surfaces that are only piece-wise smooth. Such objects possess sharp features such as corners and edges which are created when these smooth surfaces intersect. The reconstruction of these sharp features is particularly challenging as noise and sharp features are inherently ambiguous, and physical limitations in scanner resolution prevent proper sampling of such high-frequency features.

This paper proposes a denoising technique to accurately reconstruct intersections of manifolds from irregular point samples. The technique can correctly account for the anisotropic nature of sensing errors in the sampled data under the assumption that a noise model for the sensor used to acquire the points is available. The method does not assume prior availability of connectivity information, and avoids computing surface normals or meshes at intermediate steps.

* Prepared through collaborative participation in the Robotics Consortium sponsored by the U.S Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-209912.



(a) (b) (c)

Figure 1: Example of denoising a toy dataset by global fitting of an implicit degenerate polynomial (a) Input data consisting of points from two intersecting line segments corrupted with uniform Gaussian noise of std. deviation $\sigma = 0.5$ (b) Denoised data using an implicit quadratic fit with the HEIV estimator [6]. Note that the sharp feature formed by the intersection is not preserved. (c) Denoised output after imposing degeneracy constraints on fit coefficients fixes this problem.

1.1 Related Work

There have been several proposed approaches to recover geometry from noisy point samples. They may be coarsely categorized as based on computational geometry, local regression, or implicit function fitting.

In general, past approaches have often made simplifying assumptions about the data due to the ill-posed nature of the problem. (1) Methods based on classical regression typically assume that the geometry can be treated locally as a smooth surface, which is clearly a problem at surface intersections. (2) Most approaches assume the noise in the data to be isotropic and homogeneous, perhaps because they often lead to convenient closed-form analytical expressions. However, noise is almost always highly directional and dependent on the distance of the point to the sensor. This is, for example, the case with laser range scanners. Ignoring the anisotropy in the noise model typically results in a systematic bias in the surface reconstruction [6]. (3) Some methods assume the reliable availability of additional information about the geometry, such as connectivity information (meshes) and surfaces normals, and try to produce estimates of geometry that agree with this information. However, the estimation of both these quantities is error-prone. Estimation of differential quantities like surface normals and tangents is difficult in the presence of noise even for relatively smooth surfaces [7, 10], and is of course not even well-defined at intersections.

Several methods based on *computational geometry* have been developed and rigorously analyzed in the literature [3]. Many algorithms in this category come with theoretical guarantees of accuracy in the reconstruction but their applicability is largely restricted to dense low-noise datasets.

Surface estimation from noisy point samples may be posed naturally as an instance of the local *regression* problem from classical statistics. A popular non-parametric technique in this category is locally weighted regression, also known in its more general form as Savitzky-Golay filtering. As explained in [4], it adapts well to non-uniformly sampled data and exhibits less bias at boundaries. The moving least squares (MLS) technique [5] builds on this by first computing a locally approximating hyperplane and then applying a locally weighted regression procedure to the data projected to the hyperplane. The tech-

nique works well with noise but is unable to reproduce sharp features due to its implicit assumption of a single locally smooth surface.

Fleishman *et al.* [8] fit quadratic polynomials locally to data and used standard techniques from robust statistics in the fitting process. The technique relied on an initially finding low-noise local regions to obtain a reliable estimate of the quadratic fit, which may not always be feasible.

Wang *et al.* [13] proposed a more complicated procedure involving a sequence of voxelization and gap-filling, topological thinning and mesh-generation. Based on local connectivity, each voxel is classified as being at a junction, boundary and surface interior. The procedure has several points of failure, particularly at regions that are not densely sampled with respect to the voxel size.

The method presented in this paper combines the strengths of some of the previous approaches. We modify a locally weighted smoother to implicitly represent potentially multiple linear subspaces through a degenerate high-order polynomial. This allows us to explicitly model edge intersections instead of trying to fit a highly non-smooth surface. The use of a local smoother preserves the adaptability to varying sample density. By posing the regression as a constrained energy minimization problem, we can easily incorporate anisotropic error models in the data. We outline the algorithm in Section 2 and examine its behavior through several experiments in Section 3.

2 Constrained Local Regression

In this section, we describe a modified regression algorithm that will enable us to recover noise-free surfaces from noisy point cloud data, while preserving high-frequency features in the geometry. We will first consider the case of 2-D data to simplify the explanation of the main idea.

2.1 Problem definition and approach

We assume that we are given a set of points $\{\mathbf{x}_i\} \in \mathbb{R}^d$ that are assumed to be noisy observations of the positions of true points $\{\hat{\mathbf{x}}_i\} \in \mathbb{R}^d$ that lie on a locally continuous, but not necessarily smooth surface. The associated noise covariances $\Lambda_i \in \mathbf{S}_+^d$ at each point are assumed to be known, for instance, through a noise model of the sensor used to acquire the points. The points are assumed to be *irregular*, in the sense that they do not follow a known regular sampling distribution, and *unstructured* in the sense that the local connectivity of the points, such as in the form of a mesh, is not available.

Our goal will be to compute the true position $\hat{\mathbf{x}}_i$ corresponding to each observed point $\mathbf{x}_i$. The operating assumption will be that points in a local neighborhood, $\mathcal{N}(\mathbf{x}_i)$ of $\mathbf{x}_i$ may be modeled as belonging to one or more linear subspaces. This naturally suggests a maximum likelihood (or equivalently defined minimum energy) formulation of the problem, subject to the constraint that the noise-free points lie on one or more subspaces. Since the parameters of the models, number of models, as well as the association of the points to each subspaces are unknown, a popular strategy is to attempt a procedure of iterative model fitting and data association, such as Expectation-Maximization (EM). However, such iterative procedures tend to be error prone when performed with few and noisy data points, as may be expected for our problem.

Instead, we propose to model the problem as one of maximum likelihood subject to *two* types of constraints. The first type of constraint ensures that each noise-free point in

the neighborhood of interest lies on a high-order polynomial whose degree is an upper bound on the number of subspaces in that neighborhood. The second type of constraint is a function of the coefficients of the polynomial, which restricts the family of allowable polynomials to degenerate forms that can represent combinations of linear subspaces. In practice, we will sometimes relax the constraint of degeneracy to make the optimization problem more tractable at the expense of admitting a single non-linear manifold but restrict them to locally developable surfaces.

2.2 Constraints in the 2-D case

In the case of 2-D data, each local neighborhood can be modeled as as consisting of a pair of linear subspaces. Thus locally the shape may be described implicitly as a zero-level set of the equation $(\boldsymbol{\gamma}_1^{\mathrm{T}}\mathbf{x}+d_1)(\boldsymbol{\gamma}_2^{\mathrm{T}}\mathbf{x}+d_2)=0$, where $\boldsymbol{\gamma}_i \in \mathbb{R}^2$, $d_i \in \mathbb{R}$ are the parameters for each of the two linear subspaces (lines in the case of 2-D data). Note that this subsumes the case where the subspaces coincide. Expanding out the terms yields an inhomogeneous 2nd degree polynomial in 2 variables, which we will refer to as x and y corresponding to each spatial dimension.

Let us denote the coefficients of each monomial in the polynomial as given by the expression

$$\theta_1 x^2 + \theta_2 y^2 + \theta_3 xy + \theta_4 x + \theta_5 y + \theta_6 = 0. \tag{1}$$

This may be rewritten in matrix form as

$$\begin{bmatrix}\mathbf{x} & 1\end{bmatrix}\begin{bmatrix}2\theta_1 & \theta_3 & \theta_4 \\ \theta_3 & 2\theta_2 & \theta_5 \\ \theta_4 & \theta_5 & 2\theta_6\end{bmatrix}\begin{bmatrix}\mathbf{x} \\ 1\end{bmatrix} = \begin{bmatrix}\mathbf{x} & 1\end{bmatrix} A \begin{bmatrix}\mathbf{x} \\ 1\end{bmatrix} = 0. \tag{2}$$

It is a known result in algebraic geometry that a quadratic in two variables reduces to a product of two linear factors only if A is singular [1]. In fact, the case where A has only rank one corresponds to the case where the subspaces (lines) coincide.

The determinant in this case may be written explicitly to yield the equality

$$4\theta_2\theta_2\theta_6 + \theta_3\theta_4\theta_5 - (\theta_2\theta_4^2 + \theta_1\theta_5^2 + \theta_6\theta_3^2) = 0, \tag{3}$$

which can be used to constrain the solution for the θ_i's. We will denote such constraints on the coefficients of the polynomial as $\boldsymbol{\phi}(\boldsymbol{\theta}) = 0$.

2.3 Constrained optimization

Together with the constraint on coefficients we can pose the task as a constrained optimization problem defined at each point of interest $\mathbf{x} \in \{\mathbf{x}_i\}$ given by

$$\min \sum_i w_i(\mathbf{x})(\mathbf{x}_i - \hat{\mathbf{x}}_i)^{\mathrm{T}} \Lambda_i^{-1} (\mathbf{x}_i - \hat{\mathbf{x}}_i), \tag{4}$$

subject to two sets of constraints. The first set of constraints is $\boldsymbol{\theta}^{\mathrm{T}}\mathbf{v}(\hat{\mathbf{x}}_i) = 0 \ \forall i$ where $\boldsymbol{\theta} \in \mathbb{R}^m$ is the vector of monomial coefficients and $\mathbf{v}(\mathbf{x}) : \mathbb{R}^d \rightarrow \mathbb{R}^m$ is the mapping from the d-dimensional point to the monomials formed by its coordinates. For the 2-D case ($d = 2$), the number of monomial terms $m = 6$. The second constraint is that on the monomial coefficients, which is (3) in the case of 2-D data.

The weighting term $w_i(\mathbf{x})$ is used to give more importance to points closer to the point of interest $\mathbf{x}$. We can define $w_i(\mathbf{x})$ using a kernel loss function, such as a truncated Gaussian function centered at $\mathbf{x}$, so as to suitably delineate the neighborhood of interest $\mathcal{N}(\mathbf{x})$. Our implementation uses the Epanechnikov kernel $w_i(\mathbf{x}) = 1 - \|\mathbf{x}-\mathbf{x}_i\|^2/\sigma^2$ for $\|\mathbf{x}-\mathbf{x}_i\| < \sigma$ and 0 elsewhere, chosen because of its finite support and asymptotically optimal properties in related tasks such as kernel regression [12]. Here σ determines the length scale, which may be chosen differently for each $\mathbf{x}$. We comment on its selection later in Section 2.5. In what follows, we will sometimes drop the dependence on $\mathbf{x}$ in the notation for clarity, with the understanding that the optimization problem is being solved for points in a local neighborhood of *each* $\mathbf{x} \in \{\mathbf{x}_i\}$.

The standard approach to solving such a constrained optimization problem is by first forming the Lagrangian

$$\sum_i \tfrac{1}{2} w_i (\mathbf{x}_i - \hat{\mathbf{x}}_i)^{\mathrm{T}} \Lambda_i^{-1} (\mathbf{x}_i - \hat{\mathbf{x}}_i) + \sum_i \lambda_i \boldsymbol{\theta}^{\mathrm{T}} \mathbf{v}(\hat{\mathbf{x}}_i) + \boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{\theta}), \tag{5}$$

where $\{\lambda_i\}$ and $\boldsymbol{\alpha}$ are the Lagrange multipliers.

To proceed further, we linearize the equations around the current estimate of $\mathbf{x}_i$'s and $\boldsymbol{\theta}$. Let $\Delta\mathbf{x}_i = \hat{\mathbf{x}}_i - \mathbf{x}_i$ and $\Delta\boldsymbol{\theta} = \boldsymbol{\theta}_0 - \boldsymbol{\theta}$, where $\boldsymbol{\theta}_0$ is the current estimate of the true $\boldsymbol{\theta}$. To reduce notational clutter, we denote $\nabla\mathbf{v}(\mathbf{x}_i)$ by $\nabla\mathbf{v}_i$ and $\nabla\boldsymbol{\phi}(\boldsymbol{\theta}_0)$ by $\nabla\boldsymbol{\phi}_0$. This yields the equation

$$\tfrac{1}{2}\sum_i w_i \Delta\mathbf{x}_i^{\mathrm{T}} \Lambda^{-1} \Delta\mathbf{x}_i \;\; + \sum_i \lambda_i \left(\boldsymbol{\theta}_0^{\mathrm{T}} \mathbf{v}(\mathbf{x}_i) + \mathbf{v}(\mathbf{x}_i)^{\mathrm{T}} \Delta\boldsymbol{\theta} + \boldsymbol{\theta}_0^{\mathrm{T}} \nabla\mathbf{v}_i \Delta\mathbf{x}_i\right) \;\; + \boldsymbol{\alpha}^{\mathrm{T}} \left(\boldsymbol{\phi}(\boldsymbol{\theta}_0) + \nabla\boldsymbol{\phi}_0 \Delta\boldsymbol{\theta}\right) = 0. \tag{6}$$

Taking derivatives with respect to $\Delta\boldsymbol{\theta}$, $\Delta\mathbf{x}_i$ and the Lagrange multipliers yields the system of equations:

$$w_i \Lambda_i^{-1} \Delta\mathbf{x}_i + \lambda_i \boldsymbol{\theta}_0^{\mathrm{T}} \nabla\mathbf{v}_i = 0 \qquad \sum_i \lambda_i \mathbf{v}^{\mathrm{T}}(\mathbf{x}_i) + \boldsymbol{\alpha}^{\mathrm{T}} \nabla\boldsymbol{\phi}_0 = 0 \tag{7}$$

$$\boldsymbol{\theta}_0^{\mathrm{T}} \mathbf{v}(\mathbf{x}_i) + \mathbf{v}(\mathbf{x}_i)^{\mathrm{T}} \Delta\boldsymbol{\theta} + \boldsymbol{\theta}_0^{\mathrm{T}} \nabla\mathbf{v}_i \Delta\mathbf{x}_i = 0 \qquad \boldsymbol{\phi}(\boldsymbol{\theta}_0) + \nabla\boldsymbol{\phi}_0 \Delta\boldsymbol{\theta} = 0. \tag{8}$$

The solution to the above set of equations can be written as

$$\Delta\boldsymbol{\theta} = -\boldsymbol{\phi}(\boldsymbol{\theta}_0)(\nabla\boldsymbol{\phi}_0^{\mathrm{T}} \nabla\boldsymbol{\phi}_0)^{-1} \nabla\boldsymbol{\phi}_0 \tag{9}$$

$$\lambda_i = w_i (\boldsymbol{\theta}_0^{\mathrm{T}} \nabla\mathbf{v}_i^{\mathrm{T}} \Lambda_i \nabla\mathbf{v}_i \boldsymbol{\theta}_0)^{-1} \mathbf{v}(\mathbf{x}_i)^{\mathrm{T}} (\boldsymbol{\theta}_0 + \Delta\boldsymbol{\theta}) \tag{10}$$

$$\Delta\mathbf{x}_i = -\tfrac{1}{w_i} \Lambda_i \nabla\mathbf{v}_i \boldsymbol{\theta}_0 \lambda_i = -\Lambda_i \nabla\mathbf{v}_i \boldsymbol{\theta}_0 (\boldsymbol{\theta}_0^{\mathrm{T}} \nabla\mathbf{v}_i^{\mathrm{T}} \Lambda_i \nabla\mathbf{v}_i \boldsymbol{\theta}_0)^{-1} \mathbf{v}(\mathbf{x}_i)^{\mathrm{T}} (\boldsymbol{\theta}_0 + \Delta\boldsymbol{\theta}). \tag{11}$$

The above solutions to the linearized constrained optimization problem suggests an iterative technique in which a candidate initial value of $\boldsymbol{\theta}_0$ is computed and the values of $\boldsymbol{\theta}$ and the $\hat{\mathbf{x}}_i$'s are progressively modified until the constraints are satisfied. The initial value of $\boldsymbol{\theta}_0$ may be chosen as the result of an unconstrained optimization using the Fundamental Numerical Scheme (FNS) algorithm [2] or the related Heteroscedastic Errors in Variables (HEIV) method [6] based on solving a generalized eigenvalue problem.

Related formulations: At this point, we wish to comment on some related work to clarify some superficial similarities. The use of a high-order polynomial product to represent a combination of (low-order polynomial) subspaces is not new. Work by Taubin [9] fit complex 3-D curves to data, and used a high-order polynomial to represent the intersection of surfaces that formed the curve. It used an approximation to the distance function

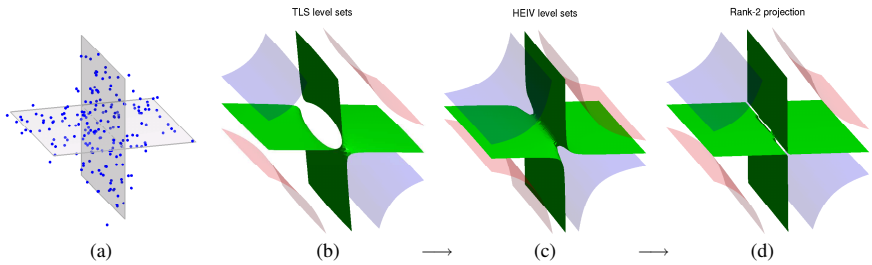


Figure 2: Illustration of sequence of optimization steps in an example of global fitting of (a) noisy observations of points lying on two planes. Level-set surfaces are shown at values 0 (green), 0.15 (red) and -0.15 (blue), and are drawn for parameters estimated with (b) TLS, which are used to initialize solution to the (c) HEIV estimate, which when subject to degeneracy constraints yields the best fit at the intersection of the planes as shown in (d).

that reduced the fitting problem to an easily solvable generalized eigenvector problem, but implicitly made the assumption of *uniform* noise covariance on the points. Vidal *et al.* [11] proposed the Generalized Principal Components Analysis (GPCA) algorithm to model combinations of linear subspaces. However, they did not consider noise in the points, and have to resort to a separate estimation procedure to compute the parameters of the individual subspaces.

In contrast, the formulation in this section explicitly incorporates a heteroscedastic noise model on the points. We use a separate constraint to capture the desired degeneracy of the polynomial as part of the optimization procedure, instead of resorting to post-processing of the result. Lastly, our focus is on *local* rather than global fitting of the data, since the data in our application cannot necessary be described globally by linear subspaces.

2.4 Constraints in the 3-D case

In the case of 3-D data, we consider the choice of model corresponding to an upper bound of 2 linear subspaces (planes) in each local neighborhood under consideration. This may be described formally as a zero-level set of the equation $(\boldsymbol{\gamma}_1^{\mathrm{T}}\mathbf{x}+d_1)(\boldsymbol{\gamma}_2^{\mathrm{T}}\mathbf{x}+d_2)=0$, where $\mathbf{x}\in\mathbb{R}^3$ and $\boldsymbol{\gamma}_i\in\mathbb{R}^3$, $d_i\in\mathbb{R}$ are the parameters for each of the two planes. Note that this again subsumes the case where the subspaces coincide. Expanding out the terms yields an inhomogeneous 2nd degree polynomial in 3 variables (denoted x, y and z).

Let us denote the coefficients of each monomial in the polynomial as given by the expression

$$\theta_1 x^2+\theta_2 y^2+\theta_3 z^2+\theta_4 xy+\theta_5 yz+\theta_6 xz+\theta_7 x+\theta_8 y+\theta_9 z+\theta_{10}=0. \tag{13}$$

This may be rewritten in matrix form as

$$\begin{bmatrix}\mathbf{x} & 1\end{bmatrix}\begin{bmatrix}2\theta_1 & \theta_4 & \theta_6 & \theta_7\\ \theta_4 & 2\theta_2 & \theta_5 & \theta_8\\ \theta_6 & \theta_5 & 2\theta_3 & \theta_9\\ \theta_7 & \theta_8 & \theta_9 & 2\theta_{10}\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ 1\end{bmatrix}=\begin{bmatrix}\mathbf{x} & 1\end{bmatrix}A\begin{bmatrix}\mathbf{x}\\ 1\end{bmatrix}=0. \tag{14}$$

Algorithm 1: DenoiseByConstrainedFitting($\{\mathbf{x}_i\}$, $\{\Lambda_i\}$)

Data: Points $X = \{\mathbf{x}_i\} \in \mathbb{R}^d$ with noise covariance $\{\Lambda_i\}$

begin

  for $\mathbf{x} \in X$ **do**

    Compute **weights** $w_i = k(\mathbf{x} - \mathbf{x}_i)$ where k is a loss function such a Gaussian

    Find the **total least squares solution** $\boldsymbol{\theta}_{\text{TLS}}$ to the unconstrained fitting problem. The least square solution is simply equal to the minimal eigenvector of the weighted covariance matrix formed by the $\mathbf{v}(\mathbf{x}_i)$'s, i.e. the minimal eigenvector of $\sum_i w_i \mathbf{v}(\mathbf{x}_i)\mathbf{v}(\mathbf{x}_i)^{\text{T}}$

    Use $\boldsymbol{\theta}_{\text{TLS}}$ to initialize the iterative solution to an **unconstrained optimization** procedure [6]. The solution to the unconstrained problem $\boldsymbol{\theta}_{\text{HEIV}}$ can be obtained through an fixed-point iteration procedure given by:

$$S(\boldsymbol{\theta}(k))\boldsymbol{\theta}(k+1) = \lambda_k C(\boldsymbol{\theta}(k))\boldsymbol{\theta}(k+1) \tag{12}$$

    where λ_k is the smallest generalized eigenvalue, and S and C are given by:

$$S(\boldsymbol{\theta}) = \sum_i \frac{A_i}{\boldsymbol{\theta}^{\text{T}} B_i \boldsymbol{\theta}} \qquad C(\boldsymbol{\theta}) = \sum_i B_i \frac{\boldsymbol{\theta}^{\text{T}} A_i \boldsymbol{\theta}}{(\boldsymbol{\theta}^{\text{T}} B_i \boldsymbol{\theta})^2}$$

    with $A_i = w_i \mathbf{v}(\mathbf{x}_i)\mathbf{v}(\mathbf{x}_i)^{\text{T}}$ and $B_i = \nabla \mathbf{v}_i^{\text{T}} \Lambda_i \nabla \mathbf{v}_i$

    Iteratively enforce the **degeneracy constraint** (3) using equations (9) and (11) (or (14) and (15) in the case of 3-D) along with the unit norm constraint $\|\boldsymbol{\theta}| = 1$ and initializing with $\boldsymbol{\theta}_{\text{HEIV}}$

  end

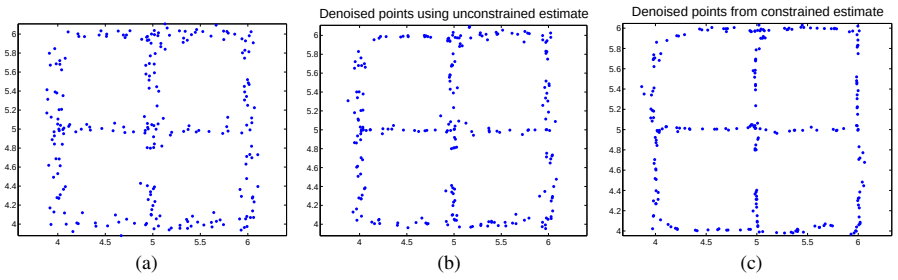
end

Following the argument in Section 2.2, it is easy to see that matrix A must be of rank 2 for the associated quadric surface to represent a pair of planes. This is equivalent to the constraints that the determinant of A as well as each of its 3×3 minors are zero. We have observed it sufficient to relax the constraints on the minors and retain the constraints only on the principal minor formed by the degree 2 coefficients, as

$$\det(B) = \det\left(\begin{bmatrix} 2\theta_1 & \theta_4 & \theta_6 \\ \theta_4 & 2\theta_2 & \theta_5 \\ \theta_6 & \theta_5 & 2\theta_3 \end{bmatrix}\right) = 0. \tag{15}$$

Geometrically, the use of this particular subset of constraints restricts the family of surfaces represented by the polynomial coefficients to the family of parallel or intersecting planes, and cylinders. Using the parameters estimated with this subset of constraints, we may then construct the matrix A, find its rank-2 approximation using its SVD decomposition, and recover the parameters of the degenerate polynomial from the rank-2 matrix.

Figure 2 illustrates the sequence of steps involved in estimating the polynomial coefficients for a synthetic dataset consists of noisy points lying on two planes intersecting at right angles. Level-set surfaces are displayed for the polynomial coefficients estimated at each step of fitting all the points. It can be seen that the TLS solution misfits the geometry, the HEIV solution tends to oversmooth the intersection (as in Figure 1 for 2-D data) and enforcing the degeneracy constraints recovers the true geometry in this example.



(a) (b) (c)

Figure 3: Example of denoising a toy dataset by *local* fitting of an implicit degenerate polynomial (a) Input data consisting of points from six line segments corrupted with spherical Gaussian noise of std. deviation $\sigma = 0.5$ (b) (b) Denoised data using an implicit quadratic fit with the HEIV estimator [6]. (c) Denoised output after imposing degeneracy constraints on coefficients.

2.5 Algorithm and Implementation

From the solution of the constrained optimization problem in the previous section, we may construct our denoising procedure as given in Algorithm 1. We draw attention to some details that influence the performance of the proposed method.

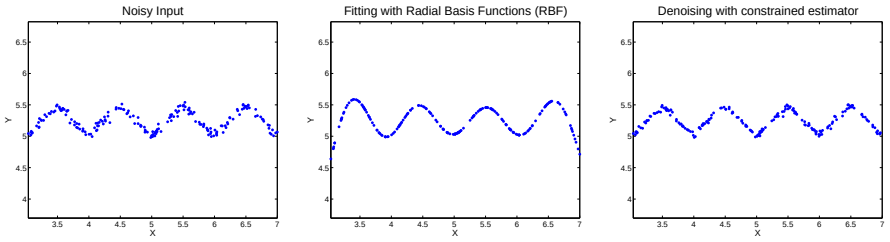
Support radius: The choice of support radius used to compute the weights w_i in the kernel function has a significant influence on the algorithm in two ways. First, the proposed method assumes an upper bound of 2 subspaces in the volume of interest, which need not be the case for any choice of support size. The chosen support radius must be one for which the modeling assumption is valid, conditional on there always existing such a choice. Secondly, even when the assumption of number of subspaces is valid, there is a tradeoff between choosing too small a radius, risking poor estimates due to the fewer number of points, or too large a radius, risking the unfavorable influence of points that do not belong to the local model.

We currently use a heuristic strategy of choosing the support radius that gives the best fit, in a maximum likelihood sense, to the corresponding neighborhood of the interest point, excluding the point itself to prevent the trivial solution of zero radius. In practice, we have observed that when the number of manifolds is under- or over-estimated, this strategy tends to reduce the support radius and show bias toward a one-manifold solution when enforcing the degeneracy constraint. However, this is an area in need of further study.

Robustness: The use of weights w_i also suggests the use of robust statistics to identify outliers to the model [8]. One strategy to identify points that have a large influence on the estimated model parameters, such as using eigenvector perturbation bounds [10] for the generalized eigenvalue problem (12) or using influence functions. In our experiments, we use a simple greedy strategy of evaluating leave-one-out fitting score and ignoring the point as an outlier if it is not a good fit with its neighbors.

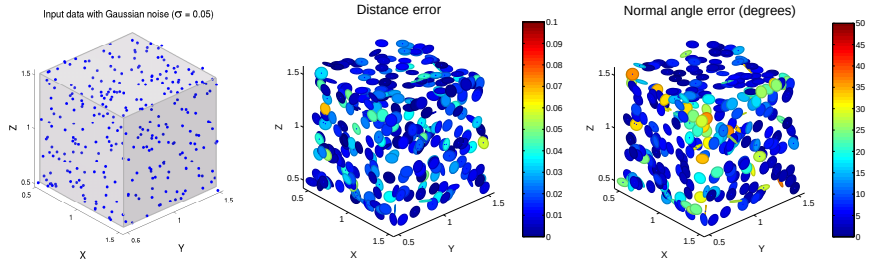
3 Experiments

We performed a series of controlled experiments of synthetic data in known configurations to evaluate the behavior of the denoising algorithm. Figure 3 shows an example where



(a) (b) (c)

Figure 4: Example of denoising samples from a triangular wave function (a) Input data corrupted with spherical Gaussian noise of std. deviation $\sigma = 0.5$ (b) Denoised data using radial basis function based smoother with Gaussian kernel. (c) Denoised output after *local* degenerate polynomial fitting.



(a) (b) (c)

Figure 5: Example of denoising samples from 3 faces of a regular cube (a) Input data corrupted with uniform noise of std. deviation $\sigma = 0.02$. Denoised points are shown with patches color coded by (b) distance error and (c) surface normal angle error.

an Epanechnikov loss function with bandwidth 0.3 was used to denoise a 2x2 square grid pattern of points. The use of a constraint enforcing degeneracy in the polynomial can be seen to preserve the intersections better than using an HEIV smoother.

Figure 4 compares the proposed fitting procedure with a standard interpolation algorithm based on radial basis functions (RBF). The RBF algorithm has two parameters [14]. The first controls the width of the Gaussian kernel which influences the locality of the smoothing. The other controls the tolerance to fitting error, i.e. a value of zero would lead to interpolation between the points, while higher values allow greater fitting error. The parameters were tuned so that the results best matched the ground-truth in the sense of least-square error. It can be seen that the proposed algorithm does a better job of preserving sharp changes in the function and is more stable at the boundary, while the RBF function tends smooths over the high curvature regions.

In Figure 5, we test the proposed algorithm on 300 noisy 3-D samples (spherical Gaussian with std. dev. 0.05) from 3 faces of a unit cube, and compared it against using the HEIV estimator from [6]. The use of the proposed estimator reduced the minimum error in normal angle over the dataset from 0.67° to 0.46° and the median distance of the points to their corresponding planes from 0.012 to 0.009 units.

4 Conclusions

In this document, we investigated the strategy of fitting local degenerate high-order polynomials to data to more faithfully represent and estimate high-frequency variations in point-sampled surfaces. The proposed strategy helps to address the inherent inability to perform differential analysis at non-manifold regions, such as intersections of curves, without actually having to estimate the parameters of component manifolds.

A current open problem is the judicious selection of the support region of the loss function. Too small a value results in a fragmented reconstruction, while the use of too large a value degrades the solution due to the influence of outliers to the implicit model. Work on an analytical solution to the optimal support radius to replace our current heuristic is in progress.

References

[1] K. Barrett. Degenerate polynomial forms. *Communications in numerical methods in engineering*, 15(5), 1999.

[2] W. Chojnacki, M. J. Brooks, A. van den Hengel, and D. Gawley. FNS, CFNS and HEIV: A unifying approach. *Journal of Mathematical Imaging and Vision*, 2005.

[3] T. K. Dey. *Curve and Surface Reconstruction*. Cambridge University Press, 2006.

[4] T. Hastie and C. Loader. Local regression: Automatic kernel carpentry. *Statistical Science*, 8(2):120–129, 1993.

[5] D. Levin. Mesh-independent surface interpolation. In *Geometric Modeling for Scientific Visualization*, pages 37–39, 2003.

[6] B. Matei and P. Meer. Estimation of nonlinear errors-in-variables models for computer vision applications. *IEEE Trans. PAMI*, 28(10):1537–1552, 2006.

[7] N. J. Mitra, A. Nguyen, and L. Guibas. Estimating surface normals in noisy point cloud data. *Journal of Computational Geometry and Applications*, 14(4), 2004.

[8] S.Fleishman, D. Cohen-Or, and C. T.Silva. Robust moving least-squares fitting with sharp features. In *Proc. ACM SIGGRAPH*, 2005.

[9] G. Taubin. An improved algorithm for algebraic curve and surface fitting. In *Intl. Conf. on Computer Vision*, 1993.

[10] R. Unnikrishnan, J.-F. Lalonde, N. Vandapel, and M. Hebert. Scale selection for the analysis of point-sampled curves. In *Proc. 3DPVT*, 2006.

[11] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (GPCA). *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(12):1945–1959, 2005.

[12] M. P. Wand and M. C. Jones. *Kernel Smoothing*. Chapman & Hall, 1994.

[13] J. Wang, M. M. Oliveira, and A. E. Kaufman. Reconstructing manifold and non-manifold surfaces from point clouds. In *Proc. IEEE Visualization*, 2005.

[14] H. Wendland. *Scattered Data Approximation*. Cambridge University Press, 2004.

Generic and Real-Time Structure from Motion

E. Mouragnon[1,2], M. Lhuillier[1], M. Dhome[1], F. Dekeyser[2] and P. Sayd[2]

[1]LASMEA UMR 6602, Université Blaise Pascal/CNRS,
63177 Aubière Cedex, France
[2]CEA, LIST, Laboratoire Calculateurs Embarqués et Image,
Boîte Courrier 65, Gif-sur-Yvette, F-91191 France ;

Abstract

We introduce a generic and incremental Structure from Motion method. By generic, we mean that the proposed method is independent of any specific camera model. During the incremental 3D reconstruction, parameters of 3D points and camera poses are refined simultaneously by a generic local bundle adjustment that minimizes an angular error between rays. This method has three main advantages: it is generic, fast and accurate. The proposed method is evaluated by experiments on real data with three kinds of calibrated cameras: stereo rig, perspective and catadioptric cameras.

1 Introduction

The automatic estimation of scene 3D structure and camera motion from an image sequence ("Structure from Motion" or SfM) has been largely studied. Different camera models are used: pinhole, fish-eye, stereo, catadioptric, multi-cameras systems, etc. A lot of **specific** algorithms (i.e. specific to a given camera model) have been successfully developed and are now well known for perspective or stereo rig models [11, 14]. The omni-directional central (catadioptric, fish-eye) or non-central (multi-cameras) systems that offer a larger field of view have also been widely explored [1, 9, 13]. It is a very interesting challenge to develop generic tools for SfM that are exploitable for any camera model. This way has recently been investigated with the introduction of **generic** camera models [7, 17]. In the generic camera model, pixels define image rays in camera coordinate system that can intersect or not in a unique point usually called "projection center". In recent work on generic SfM, camera motion can be estimated using generalization of the classical essential matrix [15, 13] given by Pless Equation [13] and minimal relative pose estimation algorithms [16].

A method is required for the refinement of 3D points and camera poses. The best solution for accuracy is bundle adjustment (BA) [18] applied on all parameters (global BA). However, it is clear that this method can not be real-time. In general, fast SfM methods or Vision-based SLAM [12, 2] (Simultaneous Localization and Mapping) are less accurate than off-line methods where an optimal solution is

calculated using global BA [18]. In this paper we present a method that makes the most of the accuracy of BA in a generic real-time application. This is possible because we developed an incremental method where not all the 3D structure is refined, but only the lastly estimated parameters at the end of the sequence. In the generic case where different cameras (pinhole, stereo, fish-eye, catadioptric...) are possible, BA is quite different from the classical one used for perspective cameras. Our generic method does not use image projections of a specific camera model but is based on back-projected rays and minimization of an angular error between rays. The first advantage is of course a high ability to change one camera model for another. The second advantage is that the method is effective if the image projection function is not explicit (as in the non-central catadioptric case) and also avoids clustering rays as in [15].

Comparison with previous works on SfM To resume, previous works are:
- generic but not real-time [15].
- real-time but not generic [2, 12], not using bundle adjustment.
- generic thanks to the use of Pless Equation [13, 15] (generalization of the epipolar constraint), but no details are given to solve this equation in common situations.
- using local bundle adjustment but not generic [10, 3], and [3] is not demonstrated in a real-time system with real world data as ours in this paper.

Contributions The first contribution of our work is a generic and real-time SfM method based on an incremental 3D reconstruction and local generic bundle adjustment where an angular error is used. The second contribution is a detailed method to solve Pless Equation (in most cases, it is not a "simple" linear problem as suggested in [13, 15]). We also compare our results with GPS ground truth and with results obtained with the most accurate (but not generic and not real-time) method available: specific global BA.

The remainder of the paper is organized as follows: Section 2 summarizes our approach and the generic camera model. The initialization method and our modified bundle adjustment are respectively explained in Section 3 and 4. Finally, experiments are presented in Section 5.

2 Overview of the Approach

2.1 Camera Model

For any pixel $\mathbf{p}$ of a generic image, the (known) calibration function f of the camera defines an optical ray $r = f(\mathbf{p})$. This projection ray is an oriented line $r = (\mathbf{s}, \mathbf{d})$ where $\mathbf{s}$ is the starting point or origin and $\mathbf{d}$ is the direction of the ray in the camera frame ($||\mathbf{d}|| = 1$). For a central camera, $\mathbf{s}$ is a unique point (camera center) whatever pixel $\mathbf{p}$. In the general case, $\mathbf{s}$ could be any point given by calibration.

2.2 Summary

The method is based on the detection and matching of interest points (Figure 1). In each frame, Harris corners [5] are detected and matched with points detected

in a previous frame by computing a Zero Normalized Cross Correlation score in a near region of interest (whatever the kind of camera). The pairs with the highest scores are selected to provide a list of corresponding point pairs between the two images. To ensure a stable estimation of the 3D, a set of frames called key frames are selected. The selection criterion is based on the number of matched points between two consecutive key frames, which must be greater than a constant.

The initialization of the geometry is provided by a method based on the resolution of Pless Equation (Section 3). Then, the algorithm is incremental. For each new video frame, (1) interest points are detected and matched with those of the last key frame (2) the camera pose of the new frame is robustly estimated (Section 3.4) (3) we check if the new frame is selected as a key frame (4) if yes, new 3D points are estimated and a local bundle adjustment (Section 4) is applied.

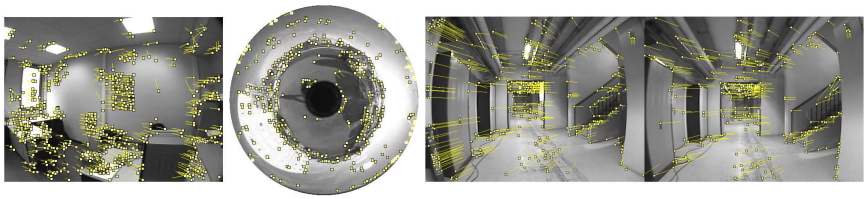


Figure 1: Feature tracks for one image of a generic camera in three cases: perspective (left), catadioptric (middle), and stereo rig (right) cameras.

3 Generic Initialization

3.1 The Pless Equation

Given a set of pixel correspondences between two images, the relative pose $(\mathtt{R}, \mathbf{t})$ of two cameras are estimated in a generic framework. For each 2D points correspondence (x_0, y_0) and (x_1, y_1) between images 0 and 1, we have a correspondence of optical rays $(\mathbf{s}_0, \mathbf{d}_0)$ and $(\mathbf{s}_1, \mathbf{d}_1)$. A ray $(\mathbf{s}, \mathbf{d})$ is defined by its Plücker coordinates $(\mathbf{q}, \mathbf{q}')$ such that $\mathbf{q} = \mathbf{d}$ and $\mathbf{q}' = \mathbf{d} \wedge \mathbf{s}$, which are convenient for this calculation. Let camera 0 be the origin of the global coordinates system and $(\mathtt{R}, \mathbf{t})$ the pose of camera 1 in this frame. The two rays must verify the generalized epipolar constraint (or Pless Equation [13])

$$\mathbf{q}_0'^{\top} \mathtt{R} \mathbf{q}_1 - \mathbf{q}_0^{\top} [\mathbf{t}]_{\times} \mathtt{R} \mathbf{q}_1 + \mathbf{q}_0^{\top} \mathtt{R} \mathbf{q}_1' = 0 \tag{1}$$

where $[\mathbf{t}]_{\times}$ is the skew symmetric cross-product matrix of the 3×1 vector $\mathbf{t}$.

We identify two cases where this equation has an infinite number of solutions. Obviously, this number is infinite if the camera is central (the 3D is recovered up to a scale). We note that Equation 1 is the usual epipolar constraint defined by the essential matrix $\mathtt{E} = [\mathbf{t}]_{\times} \mathtt{R}$ if the camera center is at the origin of the camera frame.

The second case is less obvious but it occurs in practice. In our experiments, we assume that we have only "simple" matches: all projection rays $(\mathbf{s}^i, \mathbf{d}^i)$ of a given

3D point go through a same camera center (in the local coordinate of the generic camera). In other words, we have $\mathbf{q}'_0 = \mathbf{q}_0 \wedge \mathbf{c}^0$ and $\mathbf{q}'_1 = \mathbf{q}_1 \wedge \mathbf{c}^1$ with $\mathbf{c}^0 = \mathbf{c}^1$. For a multi-camera system composed by central cameras (as the stereo rig), it means that 2D points correspondences are only made with points of the same sub-image. This is often the case in practice for two reasons: small regions of interest for reliable matching, or empty intersections between field of views of compositing cameras. If the camera motion is a pure translation ($\mathtt{R} = \mathtt{I}_3$), Equation 1 becomes $\mathbf{q}_0^\top [\mathbf{t}]_\times \mathbf{q}_1 = {\mathbf{q}'_0}^\top \mathbf{q}_1 + \mathbf{q}_0^\top \mathbf{q}'_1 = 0$ where the unknown is $\mathbf{t}$. In this context, the scale of $\mathbf{t}$ can not be estimated. We assume in this work that the camera motion is not a pure translation at the initialization step.

3.2 Solving the Pless Equation

Equation 1 is rewritten as

$$ {\mathbf{q}'_0}^\top \tilde{\mathtt{R}} \mathbf{q}_1 - \mathbf{q}_0^\top \tilde{\mathtt{E}} \mathbf{q}_1 + \mathbf{q}_0^\top \tilde{\mathtt{R}} \mathbf{q}'_1 = 0 \tag{2} $$

where the two 3×3 matrices $(\tilde{\mathtt{R}}, \tilde{\mathtt{E}})$ are the new unknowns. We store the coefficients of $(\tilde{\mathtt{R}}, \tilde{\mathtt{E}})$ in an 18×1 vector $\mathbf{x}$ and see that each value of the 4-tuple $(\mathbf{q}_0, \mathbf{q}'_0, \mathbf{q}_1, \mathbf{q}'_1)$ produces a linear equation $\mathbf{a}^\top \mathbf{x} = 0$. If we have 17 different values of this 4-tuple for each correspondence k, we have 17 equations $\mathbf{a}_k^\top \mathbf{x} = 0$. This is enough to determine $\mathbf{x}$ up to a scale factor [15]. We have built the matrix $\mathtt{A}_{17}$ containing the 17 correspondences such that $\|\mathtt{A}_{17}\mathbf{x}\| = 0$ with $\mathtt{A}_{17}^\top = [\mathbf{a}_1^\top | \mathbf{a}_2^\top | \cdots \mathbf{a}_{17}^\top]$. The resolution depends on the dimension of the $\mathtt{A}_{17}$ kernel which directly depends on the type of camera used. We determine $Ker(\mathtt{A}_{17})$ and its dimension by a Singular Value Decomposition of $\mathtt{A}_{17}$. In this paper, we have distinguished three cases: (1) central cameras with an unique optical center (2) axial cameras with collinear centers and (3) non-axial cameras.

It is not surprising that the kernel dimension of the linear system to solve is greater than one. Indeed, the linear Equation 2 has more unknowns (18 unknowns) than the non-linear Equation 1 (6 unknowns). Possible dimensions are reported in Table 1 and are justified below. Previous works [13, 15] ignored these dimensions, although a (linear) method is heavily dependent on them.

Camera	Central	Axial	Non-Axial
$dim(Ker(\mathtt{A}_{17}))$	10	4	2

Table 1: $dim(Ker(\mathtt{A}_{17}))$ depends on the kind of camera.

Central Camera For central cameras (e.g. pinhole cameras), all optical rays converge at the optical center $\mathbf{c}$. Since $\mathbf{q}'_i = \mathbf{q}_i \wedge \mathbf{c} = [-\mathbf{c}]_\times \mathbf{q}_i$, Equation 2 becomes ${\mathbf{q}_0}^\top ([\mathbf{c}]_\times \tilde{\mathtt{R}} - \tilde{\mathtt{E}} - \tilde{\mathtt{R}}[\mathbf{c}]_\times)\mathbf{q}_1 = 0$. We note that $(\tilde{\mathtt{R}}, \tilde{\mathtt{E}}) = (\tilde{\mathtt{R}}, [\mathbf{c}]_\times \tilde{\mathtt{R}} - \tilde{\mathtt{R}}[\mathbf{c}]_\times)$ is a possible solution of equation 2 for any 3×3 matrix $\tilde{\mathtt{R}}$. Such solutions are "exact": Equation 2 is exactly equal to 0 whatever $(\mathbf{q}_0, \mathbf{q}_1)$. Our "real" solution is $(\tilde{\mathtt{R}}, \tilde{\mathtt{E}}) = (\mathtt{0}, [\mathbf{t}]_\times \mathtt{R})$ if $\mathbf{c} = 0$, and it is not exact due to image noise. Thus the dimension of $Ker(\mathtt{A}_{17})$ is at least $9 + 1$. Experiments have confirmed that this dimension is 10 (up to noise). In this case, we simply solve the usual epipolar constraint constraint ${\mathbf{q}_0}^\top [\mathbf{t}]_\times \mathtt{R} \mathbf{q}_1 = 0$ as described in [6].

Axial Camera This case includes the common stereo rig of two perspective cameras. Let $\mathbf{c}_a$ and $\mathbf{c}_b$ be two different centers of the camera axis. It is not difficult to prove that "exact" solutions $(\tilde{\mathtt{R}}, \tilde{\mathtt{E}})$ are defined by

$$\tilde{\mathtt{E}} = [\mathbf{c}_a]_\times \tilde{\mathtt{R}} - \tilde{\mathtt{R}}[\mathbf{c}_a]_\times \text{ and } \tilde{\mathtt{R}} \in Vect\{\mathtt{I}_{3\times 3}, [\mathbf{c}_a - \mathbf{c}_b]_\times, (\mathbf{c}_a - \mathbf{c}_b)(\mathbf{c}_a - \mathbf{c}_b)^\top\}$$

based on our assumption of "simple" matches (Section 3.1). Our real solution is not exact due to image noise, and we note that the dimension of $Ker(A_{17})$ is at least 3+1. Experiments have confirmed that this dimension is 4.

We build a basis of 3 exact solutions $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ and a non-exact solution $\mathbf{y}$ with the singular vectors corresponding to the four smallest singular values of $\mathtt{A}_{17}$. The singular values of $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ are 0 (up to computer accuracy) and that of $\mathbf{y}$ is 0 (up to image noise). We calculate the real solution $(\tilde{\mathtt{R}}, \tilde{\mathtt{E}})$ by linear combination of $\mathbf{y}$, $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$ such that the resulting matrix $\tilde{\mathtt{R}}$ verifies $\tilde{\mathtt{R}}^\top \tilde{\mathtt{R}} = \lambda \mathtt{I}_{3\times 3}$ or $\tilde{\mathtt{E}}$ is an essential matrix. Let $\mathbf{l}$ be the vector such that $\mathbf{l}^\top = [\lambda_1 \ \lambda_2 \ \lambda_3]^\top$, and thus we denote as $\tilde{\mathtt{R}}(\mathbf{l})$ and $\tilde{\mathtt{E}}(\mathbf{l})$ the matrix $\tilde{\mathtt{R}}$ and $\tilde{\mathtt{E}}$ extracted from solution $\mathbf{y} - [\mathbf{x}_1|\mathbf{x}_2|\mathbf{x}_3]\mathbf{l}$. Using these notations, we have $\tilde{\mathtt{R}}(\mathbf{l}) = \mathtt{R}_0 - \sum_{i=1}^3 \lambda_i \mathtt{R}_i$ and $\tilde{\mathtt{E}}(\mathbf{l}) = \mathtt{E}_0 - \sum_{i=1}^3 \lambda_i \mathtt{E}_i$ with $(\mathtt{R}_i, \mathtt{E}_i)$ extracted from $\mathbf{x}_i$.

Once the basis $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ is calculated, we compute the coordinates of the solution by non-linear minimization of the function $(\lambda, \mathbf{l}) \rightarrow \|\lambda \mathtt{I}_{3\times 3} - \mathtt{R}(\mathbf{l})^\top . \mathtt{R}(\mathbf{l})\|^2$ to obtain $\mathbf{l}$ and thus $\tilde{\mathtt{E}}$. An SVD decomposition is applied to $\tilde{\mathtt{E}}$, and we obtain 4 solutions [6] for $([\mathbf{t}]_\times, \mathtt{R})$. The solution with the minimal epipolar constraint $\|\mathtt{A}_{17}\mathbf{x}\|$ is then selected. Lastly, we refine the 3D scale k by minimizing $k \rightarrow \sum_i ({\mathbf{q}'_{0i}}^\top \mathtt{R}\mathbf{q}_{1i} - \mathbf{q}_{0i}^\top k.[\mathbf{t}]_\times \mathtt{R}\mathbf{q}_{1i} + \mathbf{q}_{0i}^\top \mathtt{R}\mathbf{q}'_{1i})^2$ and perform $\mathbf{t} \leftarrow k\mathbf{t}$.

Non-Axial Camera For a non-axial camera (e.g. a multicamera system with perspective cameras such that centers are not collinear), the problem is also different. In this case, the "exact" solutions are $(\tilde{\mathtt{R}}, \tilde{\mathtt{E}}) \in Vect\{(\mathtt{I}_{3\times 3}, \mathtt{0}_{3\times 3})\}$ based on our assumption of "simple" matches (Section 3.1). The real solution is not exact due to image noise, and we see that the dimension of $Ker(A_{17})$ is at least 1+1. Experiments have confirmed that this dimension is 2. We have not yet experimented this case on real data.

3.3 Initialization with Three Views (RANSAC process)

The first step of the incremental algorithm is the 3D reconstruction of a subsequence containing the first key frames triplet $\{0, 1, 2\}$. A number of random samples are taken, each containing 17 points. For each sample, the relative pose between views 0 and 2 is computed using the abovedescribed method and matched points are triangulated. The pose of camera 1 is estimated with 3D/2D correspondences by iterative refinement minimizing the angular error defined in Section 4.2. The same error is minimized to triangulate points. Finally, the solution producing the highest number of inliers in views 0, 1, and 2 is selected from among all samples. The j-th 3D point is considered as an inlier in view i if the angular error $||\epsilon_j^i||$ is less than ϵ ($\epsilon = 0.01$ rad in our experiments).

3.4 Pose Estimates (RANSAC)

The generic pose calculation is useful for both steps of our approach (initialization and incremental process). We assume that the i-th pose $P^i = (\mathtt{R}^i, \mathbf{t}^i)$ of the camera is close to that of the i-1-th pose $P^{i-1} = (\mathtt{R}^{i-1}, \mathbf{t}^{i-1})$. P^i is estimated by iterative non-linear optimization initialized at P^{i-1} with a reduced sample of five 3D/2D correspondences, in conjunction with RANSAC. For each sample, the pose is estimated by minimizing an angular error (Section 4.2) and the overall number of counted inliers (points) includes this pose. The pose with the maximum number of inliers is then selected and another optimization is applied with all inliers.

4 Generic and Incremental Bundle Adjustment

4.1 Definitions

Bundle adjustment (BA) is the refinement of 3D points and camera poses by minimizing a cost function. The number of unknown parameters is 3 for each 3D point and 6 for each camera pose (3 for translation + 3 for rotation). Let $\mathbf{P}_j = [x_j, y_j, z_j, t_j]^\top$ be the homogeneous coordinates of the j-th point in the world frame. Let $\mathtt{R}^i$ and $\mathbf{t}^i$ be the orientation (rotation matrix) and the origin of the i-th camera frame in the world frame.

If $(\mathbf{s}^i_j, \mathbf{d}^i_j)$ is the optical ray corresponding to the observation of $\mathbf{P}_j$ through the i-th camera, the direction of the line defined by $\mathbf{s}^i_j$ and $\mathbf{P}_j$ is $\mathbf{D}^i_j = {\mathtt{R}^i}^\top[\mathtt{I}_3 \mid -\mathbf{t}^i]\mathbf{P}_j - t_j\mathbf{s}^i_j$ in the i-th camera frame. In the ideal case, directions $\mathbf{d}^i_j$ and $\mathbf{D}^i_j$ are parallel (which is equivalent to an image reprojection error of zero pixels).

4.2 Error choice

The classical approach [18, 3] consists in the minimization of a sum of square $||\epsilon^i_j||^2$ where ϵ^i_j is a specific error depending on the camera model (the reprojection error in pixels). In our case, we should minimize a generic error. We define ϵ^i_j as the angle between the directions $\mathbf{d}^i_j$ and $\mathbf{D}^i_j$ defined above.

Some experiments show that convergence of BA is bad with $\epsilon^i_j = \arccos(\mathbf{d}^i_j . \frac{\mathbf{D}^i_j}{||\mathbf{D}^i_j||})$ and satisfactory with ϵ^i_j defined as follows (a theoretical explanation of this is given in [8]). We choose $\epsilon^i_j = \pi(\mathtt{R}^i_j \mathbf{D}^i_j)$ with $\mathtt{R}^i_j$ a rotation matrix such that $\mathtt{R}^i_j \mathbf{d}^i_j = [0\ 0\ 1]^\top$ and π a function $\mathbb{R}^3 \to \mathbb{R}^2$ such that $\pi([x\ y\ z]^\top) = [\frac{x}{z}\ \frac{y}{z}]^\top$. Note that ϵ^i_j is a 2D vector whose Euclidean norm $||\epsilon^i_j||$ is equal to the tangeant of the angle between $\mathbf{d}^i_j$ and $\mathbf{D}^i_j$. The tangeant is a good approximation of the angle if it is small.

4.3 Local Generic Bundle Adjustment

In the incremental 3D reconstruction, when a new key frame I^i is selected, new matched points are triangulated. Then, a stage of optimization is carried out. It is a bundle adjustment or Levenberg-Marquardt minimization of the cost function $f^i(\mathcal{C}^i, \mathcal{P}^i)$ where $\mathcal{C}^i$ and $\mathcal{P}^i$ are respectively the generic camera parameters (extrinsic parameters of key frames) and 3D points chosen for this stage i. As it is

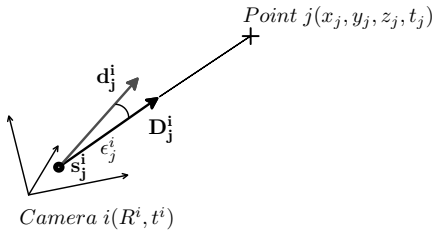


Figure 2: Angular bundle adjustment: the angle between observation ray $(\mathbf{s}_j^i, \mathbf{d}_j^i)$ and 3D ray $\mathbf{D}_j^i$ which goes from $\mathbf{s}_j^i$ to 3D point is minimized.

well known that BA is very time consuming, our idea is to reduce the number of calculated parameters and avoid redundancies in computations. In our modified BA, not all the extrinsic cameras parameters are optimized but only the n last cameras parameters. Coordinates of all 3D points seen in the last n key frames are refined including new points. To bring consistency to the incremental process and ensure that new parameters are compatible with firstly estimated ones, we take account of points reprojections in the N (with $N \geq n$) last frames (typically $n = 3$ and $N = 10$ are good values [10]). Thus, $\mathcal{C}^i$ is the camera list $\{C^{i-n+1} \ \ldots \ C^i\}$ and $\mathcal{P}^i$ contains all the 3D points projected on cameras $\mathcal{C}^i$. Cost function f^i is the sum of squared angular errors for all available observations in last key frames $C^{i-N+1} \ \ldots \ C^i$ of all 3D points in $\mathcal{P}^i$:

$$f^i(\mathcal{C}^i, \mathcal{P}^i) = \sum_{C^k \in \{C^{i-N+1} \ \ldots \ C^i\}} \ \sum_{p_j \in \mathcal{P}^i} ||\epsilon_j^k||^2.$$

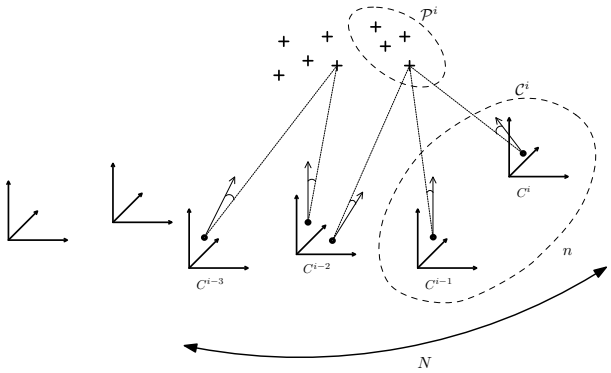


Figure 3: Local angular bundle adjustment when camera C^i is added. Only surrounded points $\mathcal{P}^i$ and cameras $\mathcal{C}^i$ parameters are optimized. Nevertheless, the minimized criterion takes account of 3D points projections in the N last images.

5 Experiments

The incremental generic 3D reconstruction method has been tested on real data with 3 different cameras: a perspective camera, a catadioptric camera and a stereo rig. Examples of frames are available in Figure 1 and sequence characteristics in Table 2. Computation performances are reported on Table 3. In the following experiments, the trajectory obtained with our generic method is compared to GPS ground truth or global specific BA result. A rigid transformation (rotation, translation and scale factor) is applied to the trajectory as described in [4] to fit with reference data. Then, a mean 3D error or 2D error in the horizontal plane can be measured between the generic and the reference trajectory.

5.1 Comparison with Ground Truth (Differential GPS)

The following results are obtained with a pinhole camera embedded on an experimental vehicle equipped with a differential GPS receiver (inch precision). The vehicle trajectory is a "S" of 88 m long (Sequence 1). The calculated motion obtained with our algorithm is compared to data given by the GPS sensor and Figure 4 shows the two trajectories registration. As GPS positions are given in a metric frame we can compare camera locations and measure positioning error in meters: mean 3D error is 1.57 m and 2D error in the horizontal plane is 1.16 m. Computation time is 2 min 32 s for the whole sequence and a mean frame-rate of 6.55 fps.

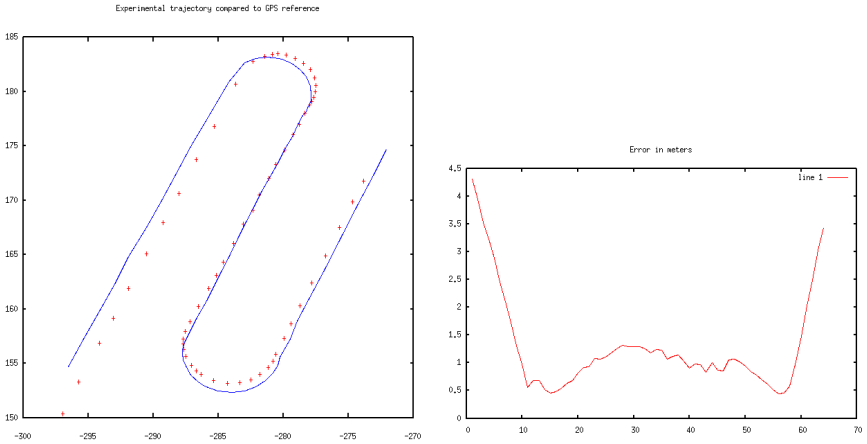


Figure 4: Left: Registration of generic vision trajectory with GPS ground truth. Continuous line represents GPS and points represent vision estimated positions. Right: 3D error (y-axis) along the trajectory (x-axis: key-frame index)

5.2 Comparison with Specific and Global Bundle Adjustment

In the two following examples, ground truth is not available. So, we compare our results with those of the best method available: a global and specific BA (all 3D parameters have been refined so as to obtain an optimal solution with a minimal reprojection error). Sequences characteristics and results are reported on Table 2.

Sequence 2 is taken in an indoor environment with a hand-held pinhole camera. A very accurate result is obtained: the mean 3D error is less than 6.5 cm for a trajectory length of (about) 15 m. The relative error is 0.45%.

Sequence 3 is taken in an outdoor environment with a hand-held catadioptric camera (the 0-360 mirror with the Sony HDR-HC1E camera visible on Figure 5, DV format). The useful part of the rectified image is contained in a circle whose diameter is 458 pixels. The accuracy is also good: the mean 3D error is less than 9.2 cm for a trajectory length of (about) 40 m. The relative error is 0.23%.

Sequence 4 is taken with a stereo rig (baseline: 40 cm) in a corridor (Figure 5). The image is composed of two sub-images of 640×480 pix. The trajectory (20 m long) is compared to results obtained with left/right camera and global BA. The mean 3D error is 2.7/8.4 cm compared to left/right camera and the relative error is 0.13/0.42%.

Sequence	Camera	#Frames	#Key frames	#3D Pts	#2D Pts	Traj. length
Sequence 1	pinhole	996	66	4808	17038	88 m
Sequence 2	pinhole	511	48	3162	11966	15 m
Sequence 3	catadioptric	1493	132	4752	18198	40 m
Sequence 4	stereo rig	303	28	3642	14189	20 m

Table 2: Characteristics of video sequences.

Camera	Image size	Detection+Matching	Frame	Key frame	Mean rate
Pinhole	512×384	0.10	0.14	0.37	**6.3 fps**
Catadioptric	464×464	0.12	0.15	0.37	**5.9 fps**
Stereo rig	1280×480	0.18	0.25	0.91	**3.3 fps**

Table 3: Computation times in *seconds* for our three cameras (detection and matching are included in Frame or Key frame times)

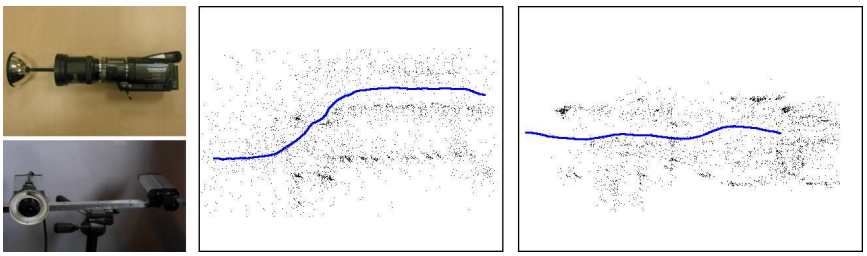


Figure 5: Left: catadioptric camera and stereo rig. Middle and right: top views of 3D reconstructions for Sequence 3 (middle) and Sequence 4 (right). Trajectory in blue and 3D points in black.

6 Conclusion

We have developped and experimented a generic method for the real-time Structure from Motion problem. We presented a complete process that starts with a generic initialization followed by an incremental 3D reconstruction of the scene and camera motion. The accuracy is brought by a local bundle adjustment minimizing an angular error. Experiments proved that it is easy to change one camera model for another, and promising results have been obtained on real data with three different kinds of cameras. Now, we are interested in experimenting our approach on more complex multi-camera systems.

References

[1] P. Chang and M. Hebert. Omni-directional structure from motion. In *Proc. of the IEEE Workshop on Omnidirectional Vision*, 2000.

[2] A. Davison. Real-time simultaneous localization and mapping with a single camera. In *Proc. of ICCV*, 2003.

[3] C. Engels, H. Stewénius, and D. Nistér. Bundle adjustment rules. In *Photogrammetric Computer Vision*, September 2006.

[4] O.D. Faugeras and M. Hebert. The representation, recognition, and locating of 3-D objects. *IJRR*, 5(3): pp 27-52, 1986.

[5] C. Harris and M. Stephens. A combined corner and edge detector. In *4th ALVEY Vision Conference*, 1988.

[6] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[7] J. Kannala and S.S. Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *PAMI*, 28(8): pp 1335-1340, 2006.

[8] M. Lhuillier. Automatic Scene Structure and Camera Motion using a Catadioptric System. *CVIU 2007*, doi: 10.1016/j.cviu.2007.05.004 (to appear).

[9] B. Micusik and T. Pajdla. Autocalibration & 3D reconstruction with non-central catadioptric cameras. In *Proc. of CVPR*, 2004.

[10] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Real time localization and 3D reconstruction. In *Proc. of CVPR*, June 2006.

[11] D. Nister. An efficient solution to the five-point relative pose problem. *PAMI*, 26(6): pp 756-777, 2004.

[12] D. Nister, O. Naroditsky, and J Bergen. Visual odometry. In *Proc. of CVPR*, 2004.

[13] R. Pless. Using many cameras as one. In *Proc. of CVPR*, 2003.

[14] M. Pollefeys, R. Koch, M. Vergauwen, and L. Van Gool. Automated reconstruction of 3D scenes from sequences of images. *Isprs Journal Of Photogrammetry And Remote Sensing*, 55(4): pp 251-267, 2000.

[15] S. Ramalingam, S. Lodha, and P. Sturm. A generic structure-from-motion framework. *CVIU*, 103(3): pp 218-228, 2006.

[16] H. Stewénius, D. Nistér, M. Oskarsson, and K. Åström. Solutions to minimal generalized relative pose problems. In *Workshop on Omnidirectional Vision*, 2005.

[17] P. Sturm and S. Ramalingam. A generic concept for camera calibration. In *Proc. of ECCV*, 2004.

[18] Bill Triggs, Philip McLauchlan, Richard Hartley, and Andrew Fitzgibbon. Bundle adjustment – A modern synthesis. In *Vision Algorithms: Theory and Practice*. 2000.

A Phase Field Model Incorporating Generic and Specific Prior Knowledge Applied to Road Network Extraction from VHR Satellite Images

Ting Peng[1,2], Ian H. Jermyn[1], Véronique Prinet[2], Josiane Zerubia[1], BaoGang Hu[2]
[1]Ariana (joint research group INRIA/I3S), INRIA, B.P. 93,
06902 Sophia Antipolis, France. Email: firstname.lastname@sophia.inria.fr
[2]LIAMA & NLPR, Institute of Automation, Chinese Academy of Sciences,
Beijing 100080, China. Email: {tpeng,prinet,hubg}@nlpr.ia.ac.cn

Abstract

We address the problem of updating road maps in dense urban areas by extracting the main road network from a very high resolution (VHR) satellite image. Our model of the region occupied by the road network in the image is innovative. It incorporates three different types of prior geometric knowledge: generic boundary smoothness constraints, equivalent to a standard active contour prior; knowledge of the geometric properties of road networks (*i.e.* that they occupy regions composed of long, low-curvature segments joined at junctions), equivalent to a higher-order active contour prior; and knowledge of the road network at an earlier date derived from GIS data, similar to other 'shape priors' in the literature. In addition, we represent the road network region as a 'phase field', which offers a number of important advantages over other region modelling frameworks. All three types of prior knowledge prove important for overcoming the complexity of geometric 'noise' in VHR images. Promising results and a comparison with several other techniques demonstrate the effectiveness of our approach.

1 Introduction

Keeping the information contained in Geographical Information Systems (GIS) up to date is crucial for many applications, for example urban planning, vehicle navigation, and environmental monitoring. The high rate of urban growth, especially in many developing countries, means that this has become an increasingly important research topic in remote sensing. Very high resolution (VHR) optical satellite images (*e.g.* QuickBird and Ikonos, and Pléiades in the near future), with sub-metric resolutions, already facilitate the updating process due to their relatively low cost, high acquisition frequency, and rich information content, but current methods, based on manual extraction, are time and labour intensive, and often surprisingly inaccurate. The development of automatic GIS updating systems is thus a necessity if the increasing demand is to be met.

In this paper, we address the updating problem for road networks, and in particular the problem of automatically updating the GIS map of main roads in Beijing using a single QuickBird panchromatic image with 0.6m resolution. Unfortunately, even restricting to

the case of road networks, the automatic updating problem is not easily solved. To extract the road network from VHR images—formally, to find the region R in the image domain Ω that contains the roads—means to ignore the wealth of 'noise' that such images contain, for instance shadows, occlusions, and entities that locally appear similar to roads. This 'noise' means that the road network cannot be identified using only the information contained in the data; a great deal of prior knowledge, in this case concerning R, must also be injected. This is particularly true in an urban environment, where the degree of 'clutter' in the image is far greater than in the peri-urban or rural cases. This prior knowledge is currently provided by the human operators who manually extract the information; the question is how to incorporate a similar quantity of prior knowledge automatically.

The knowledge needed lies at different levels of generality. The most general concerns the regularity properties of the boundary ∂R of R. These properties apply to almost any entity, not only road networks. As a consequence, this prior knowledge is included in almost all region models, *e.g.* the Ising model, and most active contour models [7]. It suffices to include a term penalizing the length of ∂R. The most specific concerns the particular road network under consideration. Seen in a more general context as 'shape modelling', this has been the subject of a number of papers in recent years, *e.g.* [3, 4, 8, 9, 14]. This type of knowledge says that the region sought must be 'close' to an exemplar region. When such an exemplar exists, *e.g.* a GIS map of the road network at an earlier date, it can significantly increase the robustness of the method. For example, Bailloeul [2] makes use of cartographic data by constraining an active contour to resemble the shape template provided by vectorized GIS building maps. Fortier et al. [5] initialize the contour using a GIS map and junctions detected in the image. The contour then corrects the position of the existing road network. Agouris et al. [1] compute a positional uncertainty for each contour point in a GIS map using fuzzy logic. An energy term measuring shape uncertainty is then used to control an active contour.

Between these two extremes is prior geometric knowledge that applies to any road network. In some ways, this is the most difficult type of prior knowledge to include in a model, mainly because the regions corresponding to road networks can possess arbitrary topology: there may be many connected components, and each connected component may contain many loops. It is a non-trivial task to combine this topological freedom with the available geometric information: road network regions are composed of long, low curvature segments of roughly constant width that join at junctions. For example, Péteri and Ranchin [11] address the problem of extracting the road network from an Ikonos satellite image in a dense urban area. They introduce geometric knowledge via a parallelism constraint on the contours representing the borders of the roads, but they avoid the topology problem by assuming that a graph of the network is given. Roads and junctions are then extracted in two steps using two different types of active contours. Rochery et al. [13] on the other hand, address the problem of road network extraction from low to medium resolution images using a modelling framework known as 'higher-order active contours'. This framework allows the inclusion of prior geometric information without necessarily constraining the topology because, rather than relying on an exemplar region, it uses long-range interactions between contour points to control region geometry. Rochery et al. [12] address road network extraction using a reformulation of HOACs as (nonlocal) phase field models. The phase field approach to region modelling, which we also use in this paper, has a number of advantages, even for the simplest models, but in particular for HOACs. Peng et al. [10] apply the work of Rochery et al. [12] to VHR images using a multiscale

data energy to deal with the complexity of such images.

In this paper, we make two main contributions with respect to this literature. In problem-specific terms, we make progress towards an automatic road map updating system for VHR images. In methodological terms, we construct a model that combines the three types of prior knowledge described above, and express them all as a nonlocal phase field prior energy. We then combine the prior model with a data energy similar in spirit to that of Peng et al. [10] but at a single scale. We test the model on a VHR image of Beijing, and compare our results to other methods in the literature.

The rest of this paper is organised as follows. In section 2, we recall the essentials of phase field methods, and then describe our model. In section 3, we discuss the algorithms used to solve the model. In section 4, we describe experimental results on VHR images. We conclude in section 5.

2 The model: prior and data energies

As outlined in section 1, our aim is to find the region R in the image domain Ω that corresponds to the main roads in the road network contained in the image. We assume that we are given a region R_0 representing the road network at an (earlier) date than the image data. Our knowledge of R is then described by a probability distribution $P(R|I,R_0,K)$, where I is the image data, and K represents all other prior knowledge we may have. From this probability distribution, we can make estimates; in particular, we can compute a MAP estimate by finding the region with maximum probability, or alternatively with minimum negative log probability, or 'energy'. Rewriting $P(R|I,R_0,K)$ using Bayes' theorem, and making a reasonable independence assumption, we can express the energy to be minimized, up to an additive constant, as

$$E(R;I,R_0) = DE_P(R,R_0) + E_D(I,R)\,, \tag{1}$$

where E_P is the prior energy (D simply weights this term), and E_D is the data energy, and K is understood.

To compute anything, one must choose a mathematical representation for R. In this paper, we use a *phase field* representation, much used in physics and first introduced to image processing in [12]. A phase field $\phi : \Omega \to \mathbb{R}$ defines a region via a threshold z: $R = \{x : \phi(x) > z\}$. Furthermore, as we will see, the phase field prior energy is so constructed that the energy-minimizing phase field ϕ_R for a fixed region satisfies $\phi_R(x) \simeq 1$ for $x \in R$ and $\phi_R(x) \simeq -1$ for $x \in \bar{R}$, where $\bar{R} = \Omega \setminus R$. As a result, the quantities $\phi_\pm = (1 \pm \phi)/2$ are approximately equal to the characteristic functions of R and $\bar{R}$. The lack of any hard constraints on ϕ, *e.g.* that it should be a distance function, is responsible for the advantages of the phase field framework over other region modelling approaches [12].

With the representation decided, we can now describe the various terms in the energy as functionals of the phase field. We will abuse notation by using the same symbol for the energy as a function of ϕ and as a function of R.

2.1 Prior energy

The prior energy E_P is itself the sum of three pieces. The first, $E_{P,0}$, is the basic phase field model, equivalent to a standard active contour with energy $\lambda_C L(\partial R) + \alpha_C A(R)$, where L is boundary length, A is region area, and λ_C and α_C are constants. This term ensures stability

of the model, boundary smoothness, and the characteristic function property mentioned earlier. The second, $E_{P,NL}$, is a nonlocal term coupling the phase field values at long distances. As shown in [12], it is equivalent to a quadratic higher-order active contour energy [13]. It introduces prior knowledge about the shapes of the regions occupied by all road networks; roughly speaking, that they are composed of long, low-curvature 'arms' of roughly constant width that join together at junctions. The third, $E_{P,GIS}$, introduces the prior knowledge specific to the road network of interest. It expresses the knowledge that R should be 'close' to R_0, which can also be described by its minimum energy phase field function ϕ_{R_0}. We now describe these pieces in more detail.

2.1.1 $E_{P,0}$ and $E_{P,NL}$

The basic phase field energy $E_{P,0}$ is given by the Ginzburg-Landau energy plus an odd parity term:

$$E_{P,0}(\phi) = \int_\Omega dx \left\{ \frac{1}{2}\nabla\phi(x)\cdot\nabla\phi(x) + W(\phi(x)) \right\}, \tag{2}$$

where the potential

$$W(y) = \lambda\left(\frac{1}{4}y^4 - \frac{1}{2}y^2\right) + \alpha\left(y - \frac{1}{3}y^3\right),$$

and λ and α are constants. For $\lambda \geqslant \alpha > 0$, W has two minima, at $y = -1$ and $y = 1$, and a maximum at $y = \alpha/\lambda$. If we ignore the gradient term, then for a fixed region R, with $z = \alpha/\lambda$, the energy-minimizing function, ϕ_R, takes value 1 inside and -1 outside R. The effect of the gradient term is to smooth this result, producing a narrow interface, centred around ∂R, that interpolates between 1 and -1.

The higher-order active contour phase field energy $E_{P,NL}$ introduces a long-range interaction between the values of ϕ at pairs of points separated by many pixels. It is given by

$$E_{P,NL}(\phi) = -\frac{\beta}{2}\iint_{\Omega^2} dx\,dx'\,\nabla\phi(x)\cdot\nabla\phi(x')\Psi((x-x')/d), \tag{3}$$

where d controls the range of the interaction. The interaction function, Ψ, is given by

$$\Psi(x) = \begin{cases} \frac{1}{2}\left(2 - |x| + \frac{1}{\pi}\sin(\pi|x|)\right) & \text{if } |x| < 2, \\ 0 & \text{else}. \end{cases}$$

In terms of ∂R, this interaction has two main effects: nearby boundary points tend to have parallel normal vectors, while those boundary points with antiparallel normal vectors increasingly repel one another as they approach closer than $2d$. These effects are responsible for the fact that the energy $E_{P,0} + E_{P,NL}$ favours regions composed of long, low curvature 'arms' of roughly constant width that join at junctions, or in other words, that it models network structures.

2.1.2 $E_{P,GIS}$

The final prior energy term, $E_{P,GIS}$, incorporates knowledge of the earlier road network, R_0. It takes the form

$$E_{P,GIS}(\phi,\phi_{R_0}) = \int_\Omega dx \left[\omega\phi_{R_0+}(x) + \bar{\omega}\phi_{R_0-}(x)\right]\left[\phi(x) - \phi_{R_0}(x)\right]^2. \tag{4}$$

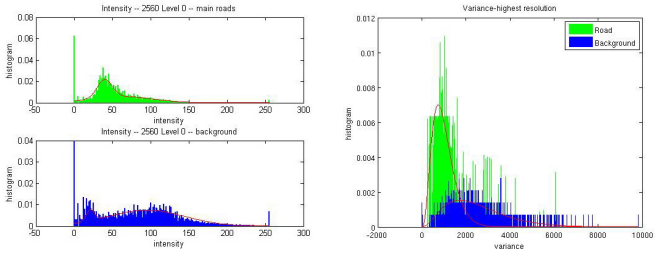


Figure 1: Histograms of the pixel intensities I on-road (top left) and off-road (bottom left), and of the variances V on-road (right, green/light grey) and off-road (right, blue/dark grey), and of the models fitted to them (solid lines).

The two terms correspond to the two components of the symmetric area difference between R and R_0: $x \in R \cap \bar{R}_0$ and $x \in \bar{R} \cap R_0$. These are separated so that they can be weighted differently by the parameters ω and $\bar{\omega}$. Because this term takes into account the exterior of R_0, it counteracts the background 'noise' appearing in the data.

2.2 Data energy

The data energy is the negative logarithm of $P(I|R,K)$. We assume that this factorizes as $P(I_R|R,K)P(I_{\bar{R}}|R,K)$, where subscripts indicate 'restricted to'. We use the same parameterized model for I_R and $I_{\bar{R}}$, the choice of model being based on a study of the image statistics. We model both the one point statistics of the image intensity, *i.e.* the histogram, and the two-point statistics, which we characterize by the variance $V(x)$ of the image in a small window around each pixel. Because of the factorization, the data energy is the sum of two pieces, one referring to R and one to $\bar{R}$ (indicated by overbars):

$$E_D(I,R) = -\int_\Omega dx \Big\{ \big[\ln P(I(x)) + \theta \ln Q(V(x))\big]\phi_+(x) + \big[\ln \bar{P}(I(x)) + \theta \ln \bar{Q}(V(x))\big]\phi_-(x) \Big\}. \quad (5)$$

Here P and $\bar{P}$ are two-component Gaussian mixture models, modelling the image intensities, while Q and $\bar{Q}$ are Gamma distributions, modelling the variances.

3 Implementation

3.1 Parameter estimation

The parameters of the Gaussian mixture and Gamma distributions are learned from the image data, using the known region R_0 to create samples of road and non-road. Note that the samples may contain errors, since R_0 does not correspond exactly to the road network in the image (see figure 2). The Gaussian mixture parameters are estimated using the EM algorithm, while the Gamma distribution parameters are estimated by least squares error minimization over the variance histograms computed in non-overlapping windows. Examples of histograms and the models fitted to them are shown in figure 1.

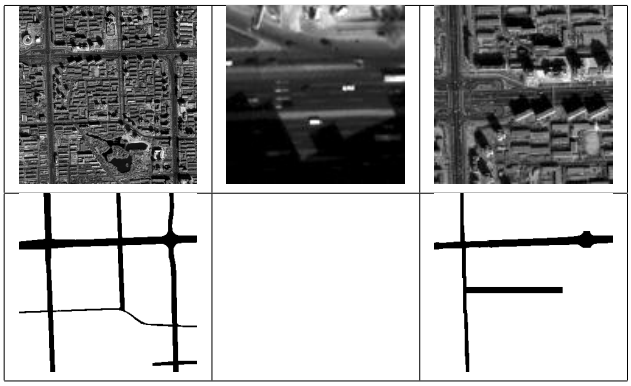


Figure 2: Top row, left to right: the QuickBird image used; a zoom on the image; a zoom on the reduced resolution image. Bottom row, left to right: ground truth, including smaller roads for comparison with other methods; deliberately 'damaged' ground truth, to simulate an earlier GIS map.

3.2 Energy minimization

The total energy functional, $E = E_D + D(E_{P,0} + E_{P,NL} + E_{P,GIS})$, is minimized with respect to ϕ using gradient descent. The functional derivative is

$$\frac{\delta E}{\delta \phi} = D\Big\{-\nabla^2\phi + \lambda(\phi^3 - \phi) + \alpha(1-\phi^2) + \beta\nabla^2\Psi * \phi + 2(\phi - \phi_{R_0})\big[\omega\phi_{R_0+} + \bar{\omega}\phi_{R_0-}\big]\Big\}$$
$$- \frac{1}{2}\Big\{\big[\ln P(I(x)) + \theta \ln Q(V(x))\big] - \big[\ln \bar{P}(I(x)) + \theta \ln \bar{Q}(V(x))\big]\Big\}, \quad (6)$$

where $*$ indicates convolution. The neutral initialization was used.

4 Experimental results

The input data I was a QuickBird panchromatic image at 0.6m resolution, as shown in figure 2, which also shows a zoom on this image. An available GIS map from a few years earlier of the road network in the zone shown in the image was used in two ways: first, to create ground truth, for which it was slightly corrected via hand segmentation; and to create an inaccurate road network region to serve as R_0. Both these are also shown in figure 2. Note that R_0 has some roads added and some roads missing. Note also that smaller roads have been kept in the ground truth; this is to allow comparison with other methods, which attempt to find all roads, not just the main road network.

We tested and evaluated the model using the original QuickBird image (0.6m/pixel), and using a lower resolution version corresponding to the scaling coefficients of a Haar wavelet decomposition of the image at level 3, *i.e.* 4.8m resolution, where level 0 is full resolution. A zoom on this image is shown in figure 2. One can see that the image at level 3 has been simplified, but is still rather complex.

The rest of this section presents the results obtained at these two resolutions, with and without GIS information, *i.e.* with and without $E_{P,GIS}$. The results are compared to those obtained using three other methods: those of Bailloeul [2], an approach based on active

Level	D	α	λ	β	d	ω	$\bar{\omega}$	θ
3	200	0.0905	3	0.02	10	0	0	0.02
0	300	0.0905	3	0.02	80	0.00033 or 0	0.0006 or 0	0.02

Table 1: Parameter values used in the experiments. Note that apart from a change in the overall weight of the prior term, and the scaling of d due to the change of resolution, they are the same for the two resolutions.

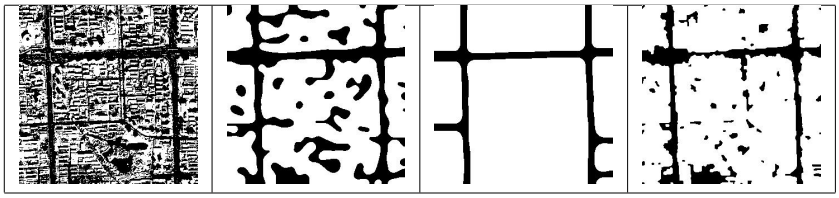


Figure 3: Experiment at reduced resolution, level 3 (320×320, road width $\simeq 12$ pixels). Three leftmost images: the thresholded phase field function at iterations 1 and 400, and at convergence, using the model without GIS information, *i.e.* without $E_{P,GIS}$. Rightmost image: for comparison, the result obtained when the nonlocal term $E_{P,NL}$ is dropped as well, leaving a model equivalent to a standard active contour. The importance of the prior geometric information carried by $E_{P,NL}$ is clear.

contours; Wang and Zhang [15], which uses classification, tracking, and morphology; Yu et al. [16], which creates a rough segmentation based on straight line density.

The parameter values for the prior energy were chosen by hand, but not freely. They are subject to a constraint that guarantees the Turing stability of the model, and a further constraint that ensures that a long bar of the desired road width is a stable configuration of the energy. The values used are given in table 1. Apart from a change in the overall weight of the prior term, and the scaling of d due to the change of resolution, they are the same for the two resolutions.

4.1 Results at reduced resolution

The leftmost three images in figure 3 show the thresholded phase field function at iterations 1 and 400 of gradient descent, and at convergence, using the model without GIS information, *i.e.* without $E_{P,GIS}$, but with the higher-order active contour prior knowledge, $E_{P,NL}$. The segmentation is very successful: the main road networks are retrieved nearly completely. The rightmost image shows the result obtained if $E_{P,NL}$ is omitted as well, leaving a model equivalent to a standard (*i.e.* not higher-order) active contour. The importance of the prior knowledge carried by the nonlocal term is clear.

By comparison, figure 4 shows the results obtained using the three methods mentioned above. The 'flexible active contour' method of Bailloeul (initially dedicated to building extraction) fails because it is not able to eliminate road sections that exist in the map but not in the image. On the other hand, the methods of Yu and Wang are able to detect the main road network and smaller roads, but, for both, the accuracy obtained in the delineation of the road boundary is poor, and the results show a great deal of noise. Some quantitative measures of the quality [6] of the results are shown in table 2.

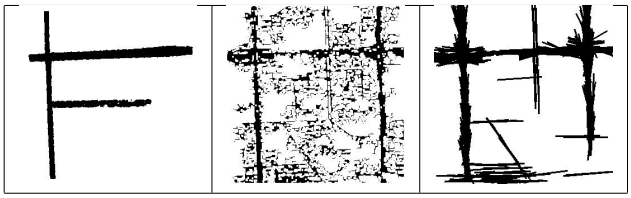


Figure 4: From left to right: the results obtained using the work of Bailloeul [2], Wang and Zhang [15], and Yu et al. [16], at reduced resolution.

Measure	Bailloeul	Wang	Yu	Our approach
TP/(TP+FN)	0.5003	0.6542	0.8240	0.7424
TP/(TP+FP)	0.7112	0.3784	0.4825	0.7382
TP/(TP+FP+FN)	0.4158	0.3153	0.4374	0.5876

Table 2: Quality measures of the different methods tested at reduced resolution (T = true, F = false, P = positive, N = negative).

4.2 Results at full resolution

The results obtained with our model, with and without GIS information, are illustrated in the leftmost two images in figure 5. The complexity of the image means that even with the prior knowledge carried by $E_{P,NL}$, without $E_{P,GIS}$ the model simply fails to retrieve the roads correctly. The addition of $E_{P,GIS}$ greatly improves the result. Its main effect is to eliminate false positives in the background, while preserving the correct segmentation of the roads themselves. To obtain this result, ω must be small, since the mistakes that may exist in the old map, should not affect the process, while $\bar{\omega}$ is somewhat bigger, because a strong constraint is needed to overcome the 'noise' in the background.

The right most image in figure 5 shows the result we obtain when we use as R_0, not the GIS map, but the result obtained at reduced resolution, level 3 (which did not use the GIS map either). This shows that in principle we can free ourselves from the need to have a GIS map available, and the full exploitation of this will be the subject of future work.

Figure 6 shows the results obtained with the methods of Bailloeul, Yu, and Wang at full resolution, while table 3 shows the corresponding quality measures.

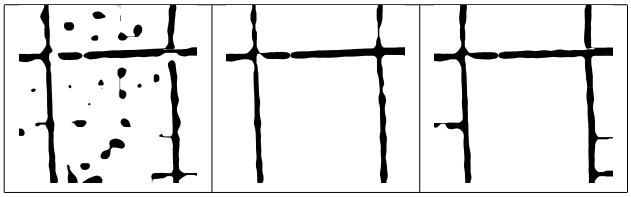


Figure 5: Experiments at full resolution, level 0 (2560×2560, road width $\simeq 96$ pixels). From left to right: the result obtained without GIS information, *i.e.* without $E_{P,GIS}$; the result obtained with GIS information, *i.e.* with $E_{P,GIS}$; the result obtained using the result obtained without $E_{P,GIS}$ at level 3 (figure 3) as a replacement for the GIS information.

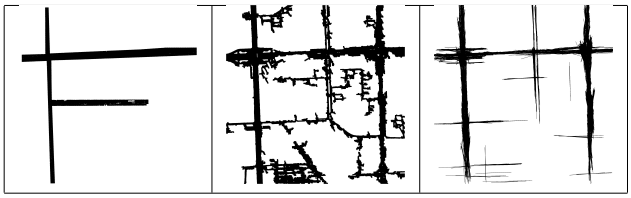


Figure 6: From left to right: the results obtained using the work of Bailloeul [2], Wang and Zhang [15], and Yu et al. [16], at full resolution.

Measure	Bailloeul	Wang	Yu	Our approach with GIS prior	Our approach with level 3 prior
TP/(TP+FN)	0.4769	0.8785	0.6706	0.6665	0.7130
TP/(TP+FP)	0.7612	0.4531	0.7836	0.9278	0.8290
TP/(TP+FP+FN)	0.4148	0.4264	0.5658	0.6336	0.6216

Table 3: Quality measures of the different methods tested at full resolution (T = true, F = false, P = positive, N = negative).

5 Conclusion

We have proposed a model for the updating of road maps in dense urban areas by extracting the main road network from a VHR satellite image. Methodologically, our model is innovative in that it incorporates three different types of prior geometric knowledge: generic knowledge about smoothness; knowledge of the geometry of road networks in general; and knowledge of the specific road network at a different date, supplied as GIS data. Our results indicate that to work at full resolution, all three types of prior knowledge are essential, due to the great complexity of VHR images. However, one can free oneself from the need for GIS data by using instead a result obtained at lower resolution, where such knowledge appears not to be necessary provided the other two types are present. Our model gives better results than three other methods in the literature, even when smaller roads, which our model is not designed to detect, are included in the ground truth.

Acknowledgments

This work was partially supported by European Union Network of Excellence MUSCLE (FP6-507752). The work of the first author is supported by an MAE/Alcatel/LIAMA grant. The authors would like to thank the Beijing Institute of Surveying and Mapping for providing the GIS data.

References

[1] P. Agouris, A. Stefanidis, and S. Gyftakis. Differential snakes for change detection in road segments. *Photogrammetric Engineering and Remote Sensing*, 67(12):1391–1399, February 2001.

[2] T. Bailloeul. *Active contours and prior knowledge for change analysis: Application to digital*

urban building map updating from optical high resolution remote sensing images. PhD thesis, CASIA and INPT, October 2005 (downloadable from http://kepler.ia.ac.cn).

[3] Y. Chen, H. Tagare, S. Thiruvenkadam, F. Huang, D. Wilson, K. Gopinath, R. Briggs, and E. Geiser. Using prior shapes in geometric active contours in a variational framework. *International Journal of Computer Vision*, 50(3):315–328, 2002.

[4] D. Cremers, F. Tischhäuser, J. Weickert, and C. Schnörr. Diffusion snakes: Introducing statistical shape knowledge into the mumford-shah functional. *International Journal of Computer Vision*, 50(3):295–313, 2002.

[5] M. F. A. Fortier, D. Ziou, C. Armenakis, and S. Wang. Automated correction and updating of road databases from high-resolution imagery. *Canadian Journal of Remote Sensing*, 27(1): 76–89, 2001.

[6] C. Heipke, H. Mayr, C. Wiedemann, and O. Jamet. Evaluation of automatic road extraction. In *Proc. International Society for Photogrammetry and Remote Sensing (ISPRS)*, volume 32, 1997.

[7] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[8] M. E. Leventon, W. E. L. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. In *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 316–322, Hilton Head Island, South Carolina, USA, 2000.

[9] N. Paragios and M. Rousson. Shape priors for level set representations. In *Proc. European Conference on Computer Vision (ECCV)*, volume 2, pages 78–92, Copenhague, Danemark, 2002.

[10] T. Peng, I. H. Jermyn, V. Prinet, J. Zerubia, and B. Hu. Urban road extraction from vhr images using a multiscale approach and a phase field model of network geometry. In *Proc. 4th IEEE GRSS/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas (URBAN)*, Paris, France, April 2007.

[11] R. Péteri and T. Ranchin. Detection and extraction of road networks from high resolution satellite images. In *Proc. IEEE International Conference on Image Processing (ICIP)*, Barcelona, Spain, September 2003.

[12] M. Rochery, I. H. Jermyn, and J. Zerubia. Phase field models and higher-order active contours. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, Beijing, China, October 2005.

[13] M. Rochery, I. H. Jermyn, and J. Zerubia. Higher-order active contours. *International Journal of Computer Vision*, 69(1):27–42, 2006.

[14] A. Srivastava, S. Joshi, W. Mio, and X. Liu. Statistical shape analysis: Clustering, learning, and testing. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(4):590–602, April 2003.

[15] R. Wang and Y. Zhang. Extraction of urban road network using quickbird pan-sharpened multispectral and panchromatic imagery by performing edge-aided post-classification. In *Proc. International Society for Photogrammetry and Remote Sensing (ISPRS)*, Quebec City, Canada, October 2003.

[16] Z. Yu, V. Prinet, C. Pan, and P. Chen. A novel two-steps strategy for automatic gis-image registration. In *Proc. IEEE International Conference on Image Processing (ICIP)*, Singapore, 2004.

Managing Particle Spread via Hybrid Particle Filter/Kernel Mean Shift Tracking

Asad Naeem[1], Tony Pridmore[1] and Steven Mills[2]

[1]School of Computer Science, University of Nottingham, Nottingham NG8 1BB, UK.

[2]Geospatial Research Centre (NZ) Ltd, Private Bag 4800, Christchurch 8140, New Zealand

Abstract

Particle filtering provides a well-developed and widely adopted approach to visual tracking. For effective tracking in real-world environments the particle set must sample widely enough that it can represent alternative target states in areas of ambiguity. It must not, however, become diffuse, spreading across the image plane rather than clustering around the object(s) of interest. A key issue in the design of particle filter-based trackers is how to manage the spread of the particle set to balance these conflicting requirements. To be computationally efficient, balance must be achieved with as small a particle set as reasonably possible. A number of hybrid particle filter/mean-shift trackers have recently been proposed. We believe that their strength lies in their ability to alternately disperse and cluster particles together, providing both a degree of balance and a reduced particle set. We present a novel hybrid of the annealed particle filter and kernel mean-shift algorithms that emphasises this behaviour. The algorithm has been applied to a wide variety of artificial and real image sequences. The method has performance and efficiency advantages over both pure kernel mean-shift and particle filtering trackers and existing hybrid algorithms

1 Introduction

The defining characteristic of the particle filter approach to visual tracking is its use of a set of discrete particles to represent multi-modal probability distributions that capture and maintain multiple hypotheses about target properties. Particle filtering is iterative. Particles are repeatedly selected, projected forwards using a motion model, dispersed by an additive random component, and evaluated against the image data. Many particle filter trackers have appeared since Blake and Isard [1] first introduced Condensation.

The ability of a set of particles to represent a wide variety of distributions is both the main strength and primary weakness of the particle filter. For effective tracking in real-world environments the particle set must sample widely enough to represent all reasonable alternatives in areas of ambiguity. It must not, however, become diffuse,

spreading across the image plane rather than clustering around the object of interest. When this happens particles tend to migrate towards local maxima in their evaluation function, becoming caught on clutter and losing track of the target. Similarly, particles should not become too focused. Though it is encouraging to see a particle set coalesce when a single, clearly distinguishable target moves across the image, the tracker should not become irreversibly locked onto a single mode.

A key issue in the design of particle filter-based trackers is how to manage the spread of the particle set to balance these conflicting requirements. The variance of the posterior is simply and elegantly maintained by the Kalman filter, but particle filters cannot assume a Gaussian, or indeed any specific, distribution. Moreover, balance must be achieved with as few particles as reasonably possible. Increasing the particle set improves representational accuracy, but adds significantly to computational overhead.

Several works have addressed aspects of this problem. Some point out that, in practice, the advantages of the particle filter approach are often lost as particles cluster, sometimes very quickly, around one target hypothesis. They focus on maintaining a wider distribution. The Annealed Particle Filter [2] uses annealing to smooth out the evaluation function, making the global maximum clearer and allowing particles to be spread further, by increasing the process noise, without becoming caught on local clutter. Vermaak et al [3] explicitly model the particle distribution as a Gaussian mixture model, forcing the resulting filter to sample an appropriate number of particles from each model component. This prevents a single, slightly more highly weighted, mode from dominating the particle distribution.

Other workers consider standard algorithms to spread the particle set too thinly across the image and concentrate effort on forcing particles to coalesce, reducing the number needed and so computational expense. The Kernel Particle Filter [4] applies a mean shift operation to the particle set to pull the centre of the particle distribution towards the target centre. This is effective, but clusters weighted particles without further reference to the image data, taking no account of the actual shape of the evaluation function between the locations sampled by the particle set. Recently, Maggio and Cavallaro [5] used a Kernel Mean Shift tracker [6] to move particles towards local maxima of the evaluation function on each iteration of Condensation.

Kernel mean shift hill climbs towards the target, minimising the distance between target and model descriptions. A spatial kernel provides some robustness to noise and partial occlusion, and the algorithm provides fast and effective tracking as long as the target object does not move further than its own diameter between frames. A number of variations on the theme have been described; a variety of colour models and similarity measures have been used and arbitrary spatial weighting [7] has been incorporated to represent objects with arbitrary or changing shapes.

Though the authors focus on the computational savings made, Maggio and Cavallaro's [5] hybrid tracker can be viewed as attempting to manage particle spread by alternately diffusing the particle set using Condensation and clustering them with Kernel Mean Shift. The algorithm shows performance advantages over both Condensation and Kernel Mean Shift, but has some drawbacks. If Condensation tends towards an incorrect local maximum the mean shift step will accelerate the process.

Recognising that the weakness of the hybrid Condensation/Mean Shift tracker lies in the particle set generated by the Condensation component, Naeem et. al. [8] propose an alternative hybrid in which Kernel Mean Shift is the dominant technology. A small number of particles are generated, in a structured fashion, to explore further when confidence in Kernel Mean Shift becomes low. Naeem et. al.'s tracker makes explicit

the iterative diffuse-cluster structure implicit in Maggio and Cavallaro's hybrid, and shows performance advantages over Condensation, Kernel Mean Shift and the Maggio and Cavallaro hybrid. The algorithm is similar in principle to the hybrid tracker of [9], which runs Kernel Mean Shift and Condensation algorithms in parallel and uses the highest confidence estimate to initialise mean shift at each time step. Naeem et. al.'s SOK tracker carries a lighter computational overhead, but requires the user to specify the conditions under which extra particles are spawned and the size of the region to be searched. This is irksome and open to error.

Here we take an alternative approach. Rather than shift control away from the particle filter component and towards the kernel mean-shift tracker we replace Condensation with a more powerful particle filter. We propose a hybrid particle filter/mean-shift tracking algorithm created by combination of the kernel Mean-Shift algorithm with Deutscher et. al.'s [2] Annealed Particle filter. We hypothesize that by smoothing out local maxima in the evaluation function the annealed particle filter will allow a greater spread in the particle set, while the Mean-Shift component will successfully pull particles back towards the true target.

The proposed Kernel Annealed Mean Shift (KAMS) tracking algorithm is presented in Section 2 and experimentally compared with Condensation [1], Kernel Mean Shift [6], annealed particle filtering [2], Maggio and Caravello's [5] condensation-based and Naeem et. al.'s [8] SOK hybrids in Section 3. Conclusions are drawn in Section 4.

2 The Kernel Annealed Mean Shift Tracker

Annealed particle filtering relies upon a series of particle weighting functions $w_0(\mathbf{Z},\mathbf{X})$ to $w_M(\mathbf{Z},\mathbf{X})$ where $\mathbf{Z}$ is a measurement vector extracted from the image and $\mathbf{X}$ is the current model state. A given weighting function w_m is obtained by raising the original weighting function $w(\mathbf{Z},\mathbf{X})$ to a power β_m, so that

$$w_m(\mathbf{Z},\mathbf{X}) = w(\mathbf{Z},\mathbf{X})^{\beta m} \tag{1}$$

where $\beta_0 = 1.0$ and $\beta_0 > \beta_1 > \beta_2 > \ldots > \beta_M$. As β_m increases, extrema in the weighting function become more pronounced. So $w_0(\mathbf{Z},\mathbf{X})$ is the raw weighting function while $w_M(\mathbf{Z},\mathbf{X})$ captures only the broad structure of the search space. In [2] $w(\mathbf{Z},\mathbf{X})$ is the sum of squared differences between the model and image data.

In annealed particle filtering each particle is evaluated at each time step using each $w_m(\mathbf{Z},\mathbf{X})$, starting with $w_M(\mathbf{Z},\mathbf{X})$ and moving to $w_0(\mathbf{Z},\mathbf{X})$. At a given time step t_k the process begins with a set of N unweighted particles

$$S_{k,M} = \{s_{k,M}^{(0)}, s_{k,M}^{(1)}, \ldots s_{k,M}^{(N)}\} \tag{2}$$

Each particle $s_{k,M}^{(i)}$ is then assigned a weight $\pi_{k,m}^{(i)}$ where

$$\pi_{k,m}^{(i)} \propto w_m(\mathbf{Z_k}, s_k \tag{3}$$

and, in the first step, $w_m(\mathbf{Z_k}, s_k^{(i)}) = w_M(\mathbf{Z_k}, s_k^{(i)})$, resulting in a set of weighted particles $S^{\pi}_{k,M}$. N particles are now drawn randomly from $S^{\pi}_{k,M}$ with replacement and used to create a set of unweighted particles for evaluation using the next weighting function

$$s_{k,M-1}^{(i)} = s_{k,M}^{(i)} + B_m \tag{4}$$

where B_m is a multi-variate Gaussian random variable with mean *0* and variance P_m. $S_{k,M-1}$ is then weighted using $w_{m-1}(\mathbf{Z_k}, s_k^{(i)})$. This is repeated until $S^{\pi}_{k,0}$ is produced

Annealing allows us to counteract the natural tendency of particle filters to cluster particles together by increasing the variance P_m, confident that the smoother weighting functions used in the early part of the annealing run will steer particles away from local extrema. Increasing the spread of the particle set, however, also increases the number of particles required to effectively sample the search area. To make explicit and accelerate the process of seeking the global maxima we apply a Kernel Mean Shift tracking step to each particle at each stage in the annealing run.

Kernel Mean Shift [6] maintains a single estimate of target position, hill climbing from the previous location estimate toward a local minimum in the Bhattacharya distance between normalized, kernel weighted color histograms representing the object model and local image data. Assuming a 3D colour histogram the Bhattacharya distance between model and candidate is:

$$bhata\,() = \sqrt{1 - \sum_{i}^{L}\sum_{j}^{L}\sum_{k}^{L} \sqrt{p(i,j,k) \times d(i,j,k)}} \tag{5}$$

where p and d are the object and the candidate models respectively. The iterative Kernel Mean Shift operation is as follows [6]:

$$x = \frac{\sum_{x=xi}^{xf}\sum_{y=yi}^{yf} x \times \sqrt{\frac{p[r_{(x,y)}, g_{(x,y)}, b_{(x,y)}]}{d[r_{(x,y)}, g_{(x,y)}, b_{(x,y)}]}}}{wt}$$

$$y = \frac{\sum_{x=xi}^{xf}\sum_{y=yi}^{yf} y \times \sqrt{\frac{p[r_{(x,y)}, g_{(x,y)}, b_{(x,y)}]}{d[r_{(x,y)}, g_{(x,y)}, b_{(x,y)}]}}}{wt} \tag{6}$$

where

$$wt = \sum_{i}^{L}\sum_{j}^{L}\sum_{k}^{L} \sqrt{\frac{p(i,j,k)}{d(i,j,k)}} \tag{7}$$

and *x* and *y* are the coordinates of the next estimate of the position of the centre of the object. In the current implementation the object and candidate are 10 x 10 x 10 bin histograms (L=10) recording RGB color values. The histogram is normalized to sum to 1. Experience has shown this to provide an effective compromise between descriptive power and ability to generalise. Though any suitable kernel could be employed, for simplicity and generality we use a linear kernel having maximum weight at the centre of the circular target area and zero weight at boundaries and beyond.

The original annealed particle filter used sum of squared difference as its base weighting function. The Kernel Mean Shift algorithm relied upon Bhattacharya

distance. To allow comparison we employ Bhattacharya distance throughout. Kernel Mean Shift is run until Bhattacharya distance either falls below a small threshold or becomes stable. Experience has shown that this usually occurs within five iterations, so a limit on the number of iterations applied can reasonably be used, if needed, to reduce computation. The final KAMS algorithm is given in Figure 1

Kernel Based Annealed Mean Shift Tracker:

1. Acquire frame at time t_k, having a set $S_{k,M}$ of N unweighted particles from the previous time step,
2. Set weighting function index m = M
3. While (m>0)
    a. Assign each particle a weight $\pi_{k,m}^{(i)}$
    b. Select N particles with replacement and add Gaussian noise: $s_{k,m-1}^{(i)} = s_{k,m}^{(i)} + B_m$
    c. Apply Kernel Mean Shift to each particle until the Bhattacharya distance between the model and image measured by the weighting function $w_{m-1}(\mathbf{Z_k}, s_{km-1}^{(i)})$ becomes stable or minimum.
    d. m = m-1

Go to 1.

Figure 1: The Kernel Annealed Mean-Shift (KAMS) tracking algorithm

3 Experimental Evaluation

3.1 Algorithms

The proposed KAMS tracker has been experimentally compared with Kernel Mean Shift [6], Condensation [1], annealed particle filtering [2] and the hybrid tracking algorithms proposed by Maggio and Caravello [5] and Naeem et. al. [8].

In Maggio and Caravello's hybrid tracker (henceforth simply "Hybrid") Condensation provides a harness into which the Kernel Mean Shift tracker outlined in section 2 is slotted. In our implementation particles are evaluated at each time step by computing the Bhattacharya distance between the object and their candidate model. Particles are then selected with probability proportional to their measurement value and projected into the next image by a constant velocity motion model. A Kernel Mean Shift tracker is initialised at each particle location and run until its associated Bhattacharya distance becomes small or constant. A limit on the number of mean shift iterations may be imposed to reduce computation without significant degradation in performance.

Naeem et. al.'s [8] Structured Octal Kernel algorithm (henceforth "SOK") is a Kernel Mean Shift tracker augmented by a backup strategy triggered when confidence in the current location estimate is low. Confidence at time *t* is given by

$$C_t = (1.0 - bhata(t)) \tag{8}$$

A user-defined threshold, *T*, is applied to *C* at each time step. If C_t is below threshold a set of eight independent Kernel Mean Shift trackers are spawned, each with the same object model as the original but at locations designed to cover a search area around the current position estimate (Figure 2). When these additional trackers have also each

converged, nine estimates of target location are available, each with an associated confidence level. The estimate with the highest confidence is selected and the process continues. In the original formulation the object model was a two-dimensional histogram of red/blue and green/blue. To allow comparison the implementation employed here uses a 10 x 10 x 10 RGB histogram.

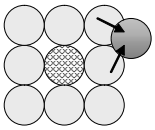


Figure 2: The SOK algorithm in operation. A hatched circle shows the primary Kernel Mean Shift tracker, light circles the secondary "particles", a dark circle the target.

3.2 Robustness

Quantitative, comparative analysis of the robustness of the proposed KAMS algorithm is achieved using McNemar's statistic [10]. McNemar's statistic is a form of chi-square test for matched paired data. Let N_{xy} give the number of times algorithm A produced result x and algorithm B produced result y, and f and s denote failure and success respectively. McNemar's statistic is then:

$$x^2 = \frac{(|N_{sf} - N_{fs}| - 1)^2}{N_{sf} + N_{fs}} \tag{9}$$

The Z score (standard score) is obtained as:

$$z = \frac{(|N_{sf} - N_{fs}| - 1)}{\sqrt{N_{sf} + N_{fs}}} \tag{10}$$

If the two algorithms give similar results then Z will tend to zero. As their results diverge, Z increases. Confidence limits can be associated with the Z value [10].

To apply McNemar's, a definition of success and failure is required. Focusing on robustness, and recognizing that any tracker will fail at some point, we consider algorithm A to have succeeded and algorithm B to have failed if algorithm A maintains tracking for a greater proportion of a given image sequence, from the same starting parameters. In effect we define success to be tracking as long as the more successful of the two trackers. McNemar's test was applied to a set of 36 assorted image sequences (available from http://www.cs.nott.ac.uk/~azn/kams_bmvc.htm) to provide quantitative comparison of the robustness of KAMS with Kernel Mean Shift, Condensation, Annealed particle filter, Hybrid and SOK. With Z scores shown in table 1, KAMS is significantly more robust than all the five algorithms with a confidence of 99.5%.

	Kams vs. Condensation	Kams vs. Mean Shift	Kams vs. Annealing	Kams vs. Hybrid	Kams vs. SOK
Z Scores	2.910428	5.126524	3.801316	4.828079	3.590662

Table 1: Z score comparisons of KAMS with the other five algorithms.

Figures 3-7 show selected frames from the results of applying the six algorithms to some of the sequences used in the McNemar's test. Figure 3 shows a tiger sprinting through dense jungle. The animal's motion is smooth, but quite fast, with frequent changes in head direction. Surrounding trees generate many partial occlusions and the dark stripes on the animal and the shadows caused by the leaves are similar, generating high levels of potentially confusing background clutter. All algorithms were manually initialised to the same point and (except SOK and Mean Shift) used 200 particles.

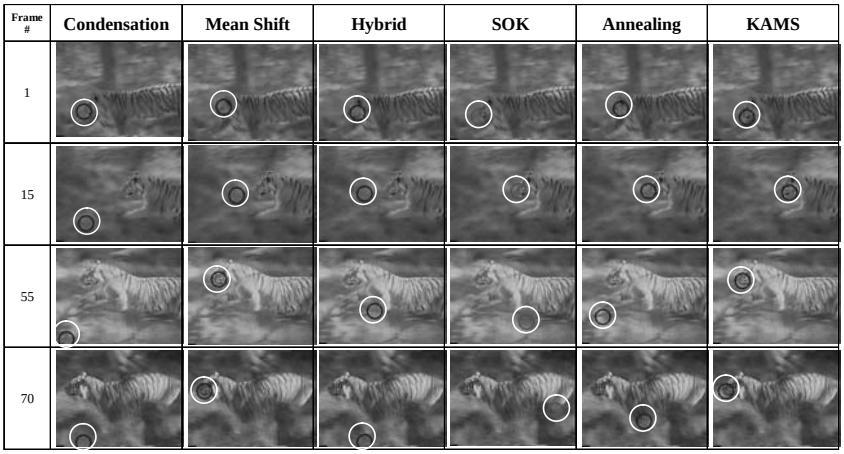


Figure 3: Six algorithms track a sprinting tiger. See supplementary material.

Condensation fails at the 15th frame; the particles are diffused and latch on to clutter resembling the tiger's head. Mean shift also fails around the 15th frame due to the high speed of movement, but latches back onto the head by chance around frame 41. Hybrid fails at frame 12 as the frequent changes in head velocity violate its motion model. SOK and the annealed particle filter fare better, and keep hold of the object until around the 50th frame, when changes in lighting conditions make clutter within their search areas appear more like the head model than the true head does.

KAMS successfully tracks to the end of the sequence. KAMS does not employ a motion model, and its combined use of particles and mean-shift allow it to use a large enough search space to keep the tiger's head within bounds, while at the same time focusing particles on the true target and so avoiding distractions. At nine times the area of the target the search area used by SOK is very large, its brute-force nature. KAMS uses more particles than SOK, but manages their spread very effectively.

Figure 4 shows the six algorithms tracking a ball moved by hand against a cluttered background. The hand moves at different velocities, sometimes partially occluding the ball. Condensation and annealed particle filtering both fail around the 5th frame as their particle sets are too dispersed and so attracted to very heavy, and very similarly coloured, clutter. Hybrid suffers the same diffusion problem, around the 55th frame. Its tight focus on the target allows the kernel mean shift to track until the 58th frame, when high target velocity throws it off. It does, however, regain the target around the 118th frame as the hand moves, by chance, underneath the wandering tracker. SOK's

dominant mean-shift tracker guides it safely through the clutter around frame 5, but it also loses track around the 60th frame. High velocity disables the mean shift component and the particle set is too widely spread to avoid clutter. Again SOK reacquires the target by chance around frame 114. KAMS tracks the ball successfully throughout the sequence. Moreover, while the other particle-based algorithms failed using 200 particles, KAMS still succeeded when its particle set was reduced to only 50 particles.

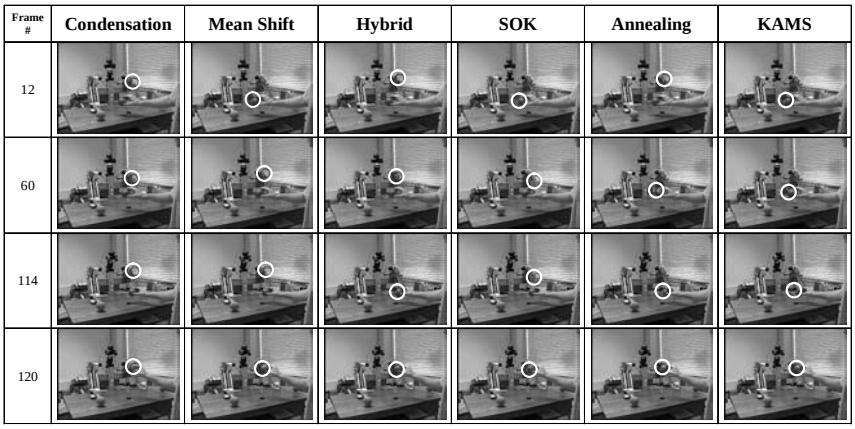


Figure 4: Six algorithms track a hand-held ball. See supplementary material.

To illustrate the key feature of the algorithm, figure 5 shows the particles generated during a single annealing run in KAMS. Each row shows the two particle sets created for a single value of *M*. The left image shows the particles after addition of Gaussian noise, the right after application of Kernel Mean Shift. Note the alternating expansion and contraction of the particle set. Note also that mean shift creates near-constant patterns of particles after only the second annealing step. Space restrictions prevent a detailed examination of the effect of varying *M*, but experience suggests that KAMS will require fewer annealing levels than pure annealed particle filtering.

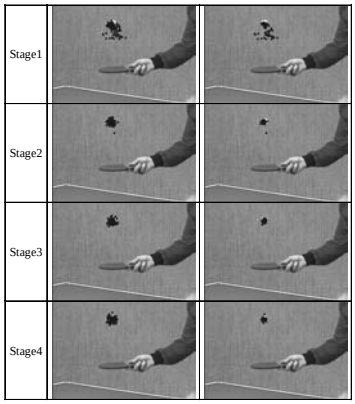


Figure 5. Dispersal and clustering of particles during an annealing run in KAMS.

Figure 6 shows a basketball player faking a pass and then passing the ball quickly from under his legs. Condensation, Mean Shift, Hybrid, SOK and pure annealing all fail around the 5th frame due to the player's high-speed and deliberately evasive movement of the ball. KAMS tracks successfully until frame 23, when the ball is totally occluded by the player's legs for 2-3 frames. Some particles briefly recquire the ball but, in the absence of a motion model, most are thrown off and the tracker loses its target.

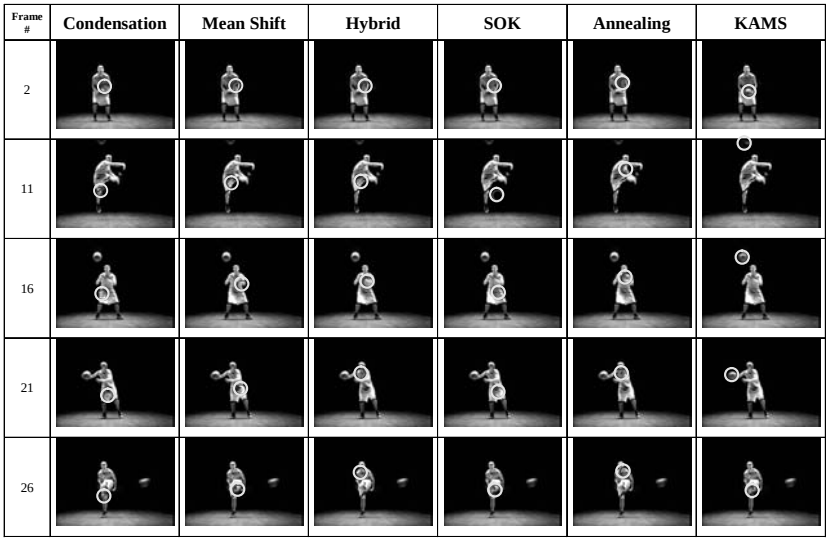
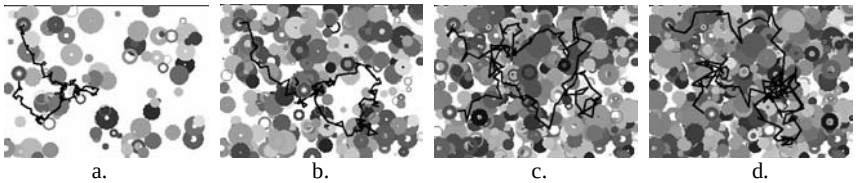


Figure 6: Six algorithms track a deliberately evasive basketball. See supplementary material.

3.2 Accuracy

Artificial sequences showing a multicolored circular target moving across a static background allow the trackers' positional estimates to be compared to ground truth in the presence of controlled amounts of measurement noise and clutter. Noise is simulated by perturbing the target's position in each frame with additive Gaussian noise. Clutter is added by randomly placing a user-defined number of similar circular objects on the otherwise white background (Figure 7). These distracting objects introduce local maxima into the evaluation function, while increased measurement noise raises the likelihood that a given tracker will come into contact with those maxima. All the artificial sequences used here consist of 140 (320x240 pixel) frames.

Only the three hybrid algorithms were included in this experiment. Hybrid completed the sequence of figure 8a with a mean error of 6.32 pixels, but failed around frame 20 when noise and clutter increased. SOK completed figures 8a. and b. with mean errors of 5.66 and 9.60 pixels, but failed thereafter. Only KAMS managed to track through all 4 sequences, with mean errors of 6.94, 18.17, 14.72 and 17.30 pixels. While the algorithms produced similar levels of accuracy (where comparable data is available), KAMS is noticeably more robust. Note also that Hybrid used 100 particles and KAMS only 40. KAMS manages its particles more efficiently and so needs significantly fewer.



a. b. c. d.

Figure 7. Artificial test sequences, a. $\sigma = 4$ with 100 background objects, b. $\sigma = 8$ with 300 objects, c. $\sigma = 12$ with 500 objects (See supplementary material), and d. $\sigma = 14$ with 600 objects. Black lines show target path, with the target displayed at either end.

4 Conclusion

Hybrid particle filter/mean shift tracking algorithms have been shown to have performance and computational advantages over their component parts. We believe the key feature of hybrid trackers to be their exploitation of the natural tension between the particle dispersal caused by the process noise of the particle filter and the clustering performed by Kernel Mean Shift. We suggest that this tension provides opportunities to better manage the spread of particles across the search space, providing higher performance with fewer particles. To test our hypothesis we have proposed a novel hybrid tracker (KAMS) that combines kernel mean-shift [6] with the annealed particle filter [2], allowing us to emphasise the iterative particle dispersal/clustering structure. The accuracies achieved by the various hybrid algorithms are comparable. The proposed algorithm, however, is significantly more robust than both previous hybrids tested and requires noticeably fewer particles than Maggio and Caravello's [5] algorithm.

References

[1] M. Isard and A. Blake, CONDENSATION – conditional density propagation for visual tracking, International. Journal of Computer Vision, 29(1) pp5-28, 1998.

[2] J. Deutscher, A. Blake, and I. Reid, Articulated body motion capture by annealed particle filtering, Proc. IEEE Conf. Computer Vision Pattern Recognition, 2000.

[3] J. Vermaak, A. Doucet and P.Perez, Maintaining multi-modality through mixture tracking, Proc. ICCV, pp 1110-1116, 2003.

[4] C. Chang and R. Ansari. Kernel particle filter for visual tracking, IEEE Signal Processing Letters, 12(3), pp. 242–245, 2005.

[5] E. Maggio and A. Cavallaro, Hybrid particle filter and Mean Shift tracker with adaptive transition model, Proc. Int. Conf. Acoustics, Speech, and Signal Processing 2005.

[6] D. Comaniciu, V. Ramesh, and P. Meer, Kernel-based object tracking, IEEE Trans. Pattern Analysis and Machine Intelligence, 25(5), pp. 564–577, 2003.

[7] A.P. Leung and S. Gong, Mean shift tracking with random sampling, Proc. BMVC 2005, pp.729-738, 2006.

[8] A. Naeem, S. Mills, and T. Pridmore, Structured Combination of Particle Filter and Kernel Mean-Shift Tracking, Proc. Int. Conf. Image and Vision Computing New Zealand, 2006.

[9] K. Deguchi, O. Kawanaka and T. Okatani, Object tracking by the mean-shift of regional colour distribution combined with the particle-filter algorithm, Proc. ICPR 2004, pp506-509, 2004.

[10] A Clark and C. Clark, Performance Characterisation in Computer Vision – A Tutorial, http://peipa.essex.ac.uk/benchmark/tutorials/essex/tutorial.pdf.

Batch Algorithm with Additional Shape Constraints for Non-Rigid Factorization

Yuan Ren Loke and Ranganath Surendra
Department of Electrical and Computer Engineering
National University of Singapore
4 Engineering Drive 3, Singapore 117576
{g0500069,elesr}@nus.edu.sg

Abstract

Recently, recovery of non-rigid structure by the factorization algorithms have received attention in the literature. The factorization algorithm decomposes the feature points over the given image sequence into motion of the camera and 3D shape bases. The non-rigid structure can be represented by the linear combination of the 3D shape bases. Although the closed-form solution of the non-rigid factorization algorithm is proven, the algorithm is sensitive to noise. In this paper, we propose a batch algorithm to recover multiple non-rigid structures from subsets of the data. Then, we introduce a set of non-linear shape constraints to optimize the recovered non-rigid structures. Synthetic data and real data were used in the experiments. The experimental results showed that the new factorization algorithm gives significant improvement than the original algorithm. With noisy data, the new algorithm is more robust and more accurate in recovering non-rigid structure.

1 Introduction

Recovering 3D structure from a sequence of images is one of typical interest topics in the computer vision community. In the past two decades, factorization algorithms have been widely applied to structure from motion (SFM) problems. It was first introduced to reconstruct rigid structure under arbitrary motion by Tomasi and Kanade [11]. Basically, the factorization algorithm for SFM decomposes the image feature tracks (*measurement matrix*) into motion of the camera and the 3D shape matrix via Singular Value Decomposition (SVD) and rank theorem. However, it is an ill-conditioned problem. Their linear transformations also yield valid motions and bases. Therefore, it is not possible to recover structure from the image sequence without some prior knowledge. Additional constraints such as orthogonality of rotation matrix are required to recover the structure.

Generally, orthographic camera model is chosen as the camera model for the factorization algorithm because it is a good approximation to the perspective camera model when the reconstructed target is far from the camera and the depth variation within the target is relatively small. [10] and [8] also proposed extended factorization algorithms for perspective and paraperspective models, respectively.

Recently, recovery of different kinds of structures such as multiple linearly moving objects [7], articulated objects [14], model based non-rigid objects [3], [1], [12], [13] are

reported. Model based non-rigid object recovery is attractive because many interesting non-rigid objects in nature such as human face can be represented by models. Reconstructing 3D human faces is very useful in face recognition. Compared to 2D face images, 3D face are invariant to pose changes. The pose changes significantly affect the performance of face recognition algorithms. Therefore, we can use non-rigid factorization to decompose the pose and deformation of the non-rigid structure from a image sequence.

To model the deformation of these non-rigid objects, the weighted combination of basis shapes has been applied in non-rigid SFM [3]. Using this model, Jing Xiao et al. [13] showed a closed-form solution for non-rigid SFM with rotation constraints and basis constraints. The solution is exact only when the data is noise free. The method does not work satisfactorily with noisy data [2].

In this paper, a batch algorithm and a non-linear shape constraint optimization are proposed to improve the existing closed-form solution under noisy environments. The batch algorithm partitions the matrix and recovers 3D structures from each partition separately. Then we apply the optimization algorithm to refine the closed-form solution of each partition based on shape constraints. Qualitative and quantitative evaluation showed that the new algorithm gives more robust and more accurate results compared to the original factorization method for both rigid and non-rigid structure.

2 Overview of Factorization Algorithm for Non-rigid SFM

Here the camera model is assumed to be the weak perspective projection model. We also assumed that the motion is non-degenerate. Let the 2D image coordinates of P feature points over F frames denoted as $\mathbf{W} = \{\mathbf{w}_{fp} = (u_{fp}, v_{fp}) | f = 1, \ldots, F, p = 1, \ldots, P\}$, the $2F \times P$ *measurement matrix*:

$$\mathbf{W} = \begin{bmatrix} u_{11} & \ldots & u_{1P} \\ v_{11} & \ldots & v_{1P} \\ \vdots & u_{fp} & \vdots \\ \vdots & v_{fp} & \vdots \\ u_{F1} & \ldots & u_{FP} \\ v_{F1} & \ldots & v_{FP} \end{bmatrix} \tag{1}$$

The camera projection matrix is written as:

$$\mathbf{R}_f = \begin{bmatrix} r_{f1} & r_{f2} & r_{f3} \\ r_{f4} & r_{f5} & r_{f6} \end{bmatrix} \qquad f \in \{1, \ldots, F\} \tag{2}$$

The non-rigid structure is represented by a linear combination of K 3D shape bases. Let $\mathbf{S}_f = \{\mathbf{s}_{fp} = (x_p, y_p, z_p) | p = 1, \ldots, P\}$ denote the 3D non-rigid structure of the f^{th} frame. Let $\mathbf{B} = \{\mathbf{b}_k = (x_{kp}, y_{kp}, z_{kp}) | k = 1, \ldots, K, p = 1, \ldots, P\}$ denote as the 3D shape bases. Then, the 3D non-rigid structure in each frame can be represented as:

$$\mathbf{S}_f = \sum_{k=1}^{K} c_{fk} \mathbf{b}_k \qquad f \in \{1, \ldots, F\} \tag{3}$$

where c_{fk} are the weights. Then, $\mathbf{W} = \mathbf{MB} + \mathbf{T}$ where $\mathbf{M}$ is a $2F \times 3K$ motion matrix, $\mathbf{B}$ is a $3K \times P$ 3D structure matrix and $\mathbf{T}$ is a $2F \times 1$ translation vector. When K=1, the structure is rigid. The motion matrix is the product of the weighting coefficients and the corresponding camera projection matrices. We can write this as

$$\mathbf{M} = \begin{bmatrix} c_{11}\mathbf{R}_1 & \dots & c_{1K}\mathbf{R}_1 \\ \vdots & c_{fk}\mathbf{R}_f & \vdots \\ c_{F1}\mathbf{R}_F & \dots & c_{FK}\mathbf{R}_F \end{bmatrix} \tag{4}$$

The translation vector can be obtained by computing the mean of the P feature points. The *registered measurement matrix*, $\hat{\mathbf{W}}$ is given by subtracting $\mathbf{T}$ from $\mathbf{W}$. The world origin now is placed at the centroid of the feature points, i.e.

$$\frac{1}{P}\sum_{p=1}^{P} \mathbf{w}_{fp} \qquad \forall f \in \{1,\dots,F\} \tag{5}$$

When the data is noiseless, the rank of $\hat{\mathbf{W}}$ is $3K$. Applying SVD, $\hat{\mathbf{W}}$ can be decomposed into a motion matrix, $\hat{\mathbf{M}}$ and a 3D basis matrix, $\hat{\mathbf{B}}$. However, it is only up to an arbitrary $3K \times 3K$ invertible transformation, $\mathbf{G}$. The exact motion matrix, $\mathbf{M}$ and 3D basis matrix, $\mathbf{B}$ can be written as:

$$\begin{aligned} \mathbf{M} &= \hat{\mathbf{M}} \cdot \mathbf{G} \\ \mathbf{B} &= \mathbf{G}^{-1} \cdot \hat{\mathbf{B}} \end{aligned} \tag{6}$$

The *corrective transformation matrix*, $\mathbf{G}$ is compound of K $3K \times 3$ matrix, $\mathbf{G}_k$. Then, $\mathbf{Q}_k = \mathbf{G}_k\mathbf{G}_k^T$. Computing the $\mathbf{Q}_k$ requires additional constraints. We have

$$\hat{\mathbf{M}}\mathbf{Q}_k\hat{\mathbf{M}}^T = \begin{bmatrix} c_{1k}\mathbf{R}_1 \\ \vdots \\ c_{1k}\mathbf{R}_F \end{bmatrix} \begin{bmatrix} c_{1k}\mathbf{R}_1 & \dots & c_{1k}\mathbf{R}_F \end{bmatrix} \tag{7}$$

Since rotation matrices are orthonormal, we have $\mathbf{R}_i\mathbf{R}_i^T = \mathbf{I}_{2\times 2}$. In [13], it was showed that using only these rotation constraints is insufficient to uniquely determine $\mathbf{Q}_k$. Thus, they also assume the first K images to be basis images. The corresponding weighting coefficients are then

$$c_{ij} = \begin{cases} 1 & \text{when } i = j \\ 0 & \text{when } i \neq j \end{cases} \tag{8}$$

We can now obtain a closed-form solution for each $\mathbf{Q}_k$ by optimizing the rotation and basis constraints. For the details of proof, the reader is referred to [13].

3 Batch Algorithm Using Matrix Partitioning

In practice, a large number of frames from video sequence are available, and using all the frames in SVD algorithm to minimize $\|\mathbf{W} - \mathbf{MB}\|_F$ may bring no advantage, firstly, because there is a large amount of redundancy in the video frames (this is just increasing the computational cost). and secondly, minimizing $\|\mathbf{W} - \mathbf{MB}\|_F$ does not guarantee that

the recovered structure is optimal. The solutions of the motion matrix $\mathbf{M}$ and the bases $\mathbf{B}$ also depend on the constraints we used on solving the corrective transformation matrix, $\mathbf{G}$.

Hence, we introduce a batch algorithm where a registered measurement matrix is partitioned into N submatrices. Then, the closed-form solution method is applied to each separately. This yields N estimates instead of a single estimate for the structures from a large number of frames. We hence expect that the proposed algorithm will improve the confidence in the result. We then propose to use these in a shape constrained non-linear optimization technique to find the best shape estimate.

Let $\Omega_i \subset \{1,\ldots,F\}$, $i = 1,\ldots,N$ be a subset of frame indexes. Then, let $\mathbf{W}_{\Omega_i} = \{(u_{fp}, v_{fp}) | f \in \Omega_i, p = 1,\ldots,P\}$ denote a row subspace of the matrix, where $|\Omega_i| \geq \max(\frac{K^2+K}{2}, 3K)$. The union of all subsets Ω_i contains all the elements of $\{1,\ldots,F\}$. All subsets are disjoint. Hence, the information in every frame is used for recovery of the structure.

Here, we assume that K is known. The set of K basis images which give the smallest condition number is the set of the most independent basis images. Thus, we can selected them as the K basis images.

Since the rank of $\hat{\mathbf{W}}_{\Omega_i}$ has to be at least $3K$, the number of frames in each partition can be determined in such a way that reasonable amount of the energy of $\hat{\mathbf{W}}_{\Omega_i}$ remains in the first $3K$ eigen-subspaces. Then each $\hat{\mathbf{W}}_{\Omega_i}$ can be decomposed by the non-rigid factorization algorithm discussed in Section 2 as:

$$\mathbf{W}_{\Omega_i} = \mathbf{M}_{\Omega_i}\mathbf{B}_i \qquad i = 1,\ldots,N \tag{9}$$

The recovered structures are exact for noiseless data.

When $K = 1$ (rigid case), the motion matrix $\mathbf{M}$ and $\mathbf{B}$ are simplified as rotation matrix $\mathbf{R}$ and the rigid structure matrix $\mathbf{S}$. When $K \geq 2$ (non-rigid case), we do not only need to recover the bases $\mathbf{B}$, but also the weighting coefficients in the motion matrix $\mathbf{M}$ for recovering the 3D structure. $\mathbf{M}$ can be obtained as

$$\mathbf{M}_i = \mathbf{W}\mathbf{B}_i^{+} \qquad i = 1,\ldots,N \tag{10}$$

where $\mathbf{B}_i^{+}$ is the pseudo-inverse of $\mathbf{B}_i$. Since the rotation matrix $\mathbf{R}_f$ is orthonormal, $||\mathbf{R}_f|| = 1$. The corresponding coefficients for each frame can be easily extracted out from motion matrix.

Let N sets of the estimated structures of the f^{th} frame denote as $\{\tilde{\mathbf{S}}_f\}_i$. Given the 3D shape bases $\mathbf{B}_i$ and the corresponding coefficients, each recovered structure can be computed by (3). Since each set of the recovered structures, $\{\tilde{\mathbf{S}}_f\}_i$ is independently estimated from the corresponding $\mathbf{W}_{\Omega_i}$, the reference coordinate systems of each two sets of the recovered structures are different up to a 3×3 orthonormal transformation. The orthonormal transformation can be obtained by applying Procrustes method.

4 Non-linear Shape Constraint Optimization

Here, we introduce an objective function which is optimized to enforce non-linear shape constraints and estimate the best recovered structure $\mathbf{S}_f$ from the set of estimated struc-

tures $\{\tilde{\mathbf{S}}_f\}_i$ from each partition. It is given as:

$$\min \sum_{n=1}^{N} \sum_{i=1}^{P} \sum_{j=1}^{P} \|s_{fi}\tilde{s}_{fj}^T - \tilde{s}_{fin}\tilde{s}_{fjn}^T\|^2 \quad \forall f \in \{1,\ldots,F\} \tag{11}$$

where N is the number of partitions. This optimization minimizes the inner products of every two feature points. In other words, we are optimizing the errors in the lengths and the mutual angles of the feature points, so we named it *metric optimization*. The metric optimization plays a role in structure refinement of the factorization method. A general-purpose quasi-Newton method [4],[5],[6],[9] is used to find the optimum solution of (11).

The initialization is critical for non-linear optimization problems. To avoid the solution of the metric optimization from being trapped at an unsuitable local minimum, we choose the least mean square of $\{\tilde{\mathbf{S}}_f\}_i$ as the initial value for the metric optimization. In the experiments discussed in the following section, we show that the metric optimization gives more robust and better results than the original algorithm.

Our proposed algorithm is summarized as follows:

1. Partition the measurement matrix $\mathbf{W}$ into N submatrices.
2. Choose the K basis images from each subset based on their condition numbers. The set of the K basis images with the smallest condition number is the set of the most independent basis images.
3. Apply non-rigid factorization algorithm proposed by Jing Xiao et al. [13]
4. Extract the weight c_{fk} from the motion matrix $\mathbf{M}$.
5. Compute the structures by Eq. (3).
6. Optimize the estimated structures obtained in Step 5 by the objective function in Eq. (11).

5 Experiments

We evaluated the proposed factorization algorithm with metric optimization quantitatively and qualitatively on synthetic data and facial expression images, respectively. In the quantitative evaluation, our approach was applied on rigid and non-rigid synthetic data sets. In the qualitative evaluation, a set of human face expressions was used to examine the performance of our approach. The results are presented below.

5.1 Quantitative Evaluation on Synthetic Data

In this section, three approaches were evaluated on synthetic data. The first approach is Jing Xiao et al's [13] non-rigid factorization algorithm. The second approach applies the batch algorithm to estimate the 3D structures from each partition. The optimum structure is the mean of the estimated 3D structures which was the smallest mean square distance to the 3D estimated structures. The third approach is the batch algorithm with metric optimization. Two experiments were carried out to examine the performance of the algorithms.

In the first experiment, 15 rigid object datasets with Gaussian white noise were generated. The strength level of noise is defined as $\frac{\|\mathbf{noise}\|}{\|\mathbf{W}\|}$. Each dataset has 50 3D feature points and 100 frames with random projection matrices. A 200×50 measurement matrix $\mathbf{W}$ represented the image feature tracks. In the second experiment, 5 non-rigid object datasets formed by 3 shapes bases were generated. Each dataset has 25 3D feature points and 203 random projection matrices. A 406×25 measurement matrix $\mathbf{W}$ represented the image feature tracks. For the non-rigid dataset, Gaussian white noise was added at strength levels of 5%, 10% and 20% to evaluate the performance of the algorithms.

To make the experiments comparable, all the synthetic datasets were partitioned into 10 subsets and batch algorithm of section 3 was applied. For rigid case, each subset contains 50 3D feature points and 10 random projection matrices. For non-rigid case, each subset contains 25 3D feature points and 13 random projection matrices (3 basis images + 10 non-basis images). They formed 10 smaller measurement matrices $\mathbf{W}_i$. Then we applied non-rigid factorization algorithm on each $\mathbf{W}_i$ to recover 3D structures. Metric optimization is applied on these 3D estimated structures by quasi-Newton optimization algorithm.

For rigid case, the relative error measurement, $\frac{1}{P}\sum_{p=1}^{P}\frac{\|\mathbf{b}_p-\mathbf{b}_p^{truth}\|}{\|\mathbf{b}_p^{truth}\|}$ was evaluated for examining the performance of our approach. For non-rigid case, the mean of the relative errors between the optimal structure and the ground truth, $\frac{1}{PF}\sum_{p=1}^{P}\sum_{f=1}^{F}\frac{\|\mathbf{s}_p-\mathbf{s}_p^{truth}\|}{\|\mathbf{s}_p^{truth}\|}$, was used instead. The results are shown in Figure 1. From the Figure 1, the relative error of the proposed algorithm is significantly lower than [13] factorization algorithm. The variance of the error is also small, showing that the method is more stable and robust than the original factorization algorithm.

5.2 Qualitative Evaluation on Facial Expressions

Recognizing facial expressions is one of the current challenging problems. Thus, we are motivated to evaluate our approach with facial expressions. In this experiment, a 3D face model with four different expressions captured from 3D Facial Expression Database [15] at the State University of New York was used to examine the qualitative performance of our proposed approach. The four expressions are happy, neutral, sad and surprise. First, we manually selected 68 feature points on the 3D models. Then, the 3D models were rotated about x-axis from -10° to -10° in 2° steps, about y-axis from -20° to -20° in 1° steps and about z-axis from -10° to -10° in 2° steps. In each step, we generated an image of the 3D model. Therefore, we have 4961 images for each expression. Some images with different expressions are shown in Figure 2. The ground truth of the 3D feature points of each expression is shown in Figure 3.

In this experiment, four different levels of Gaussian white noise were added to $\mathbf{W}$, with strength levels of 0%, 5% and 10%. Then, $\mathbf{W}$ is partitioned into 41 subsets for the batch algorithm and each subset is applied factorization with metric optimization. The results are showed in Figure 4, Figure 5 and Figure 6, respectively.

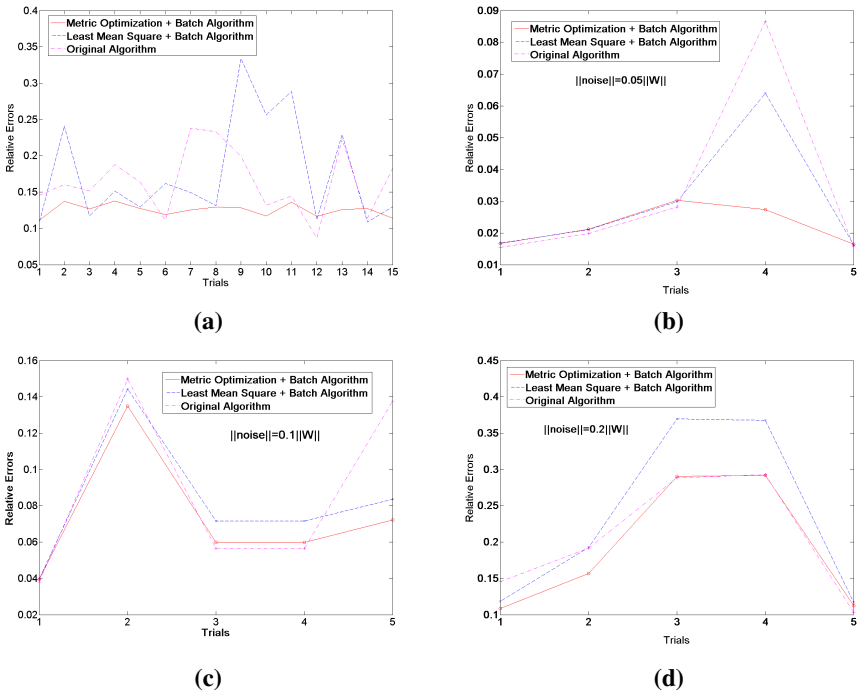


Figure 1: Relative errors of the three different appoaches of the factorization algorithms on rigid synthetic data (*a*) and non-rigid data under different levels of Gaussian white noise (*b*, *c* and *d*).

6 Discussion and Conclusion

Our approach is an extension of the non-rigid factorization algorithm proposed by Xiao et al. [13]. In this paper, we proposed a batch algorithm which uses partitions of the measurement matrix and a metric optimization that recovers the optimized 3D structures based on nonlinear shape constraints. The batch algorithm allows the system to process the data in parallel because the factorization algorithm can be applied on each submatrix separately. Thus, it is suitable for real-time applications such as surveillance and biometric authentication systems. The algorithm does not require to repeatedly compute the factorization algorithm with the whole measurement matrix every time the new data are added. The computation becomes more effective by using our proposed approach.

The metric optimization is another significant contribution in this paper. We introduced the metric optimization to refine the recovered 3D structures by using the new shape constraints. The experiments showed that our approach is more accurate and robust than the existing factorization algorithm for both rigid and non-rigid objects under different strength levels of Gaussian white noise.

References

[1] Matthew Brand. Morphable 3d models from video. *In Proc. Int. Conf. Computer Vision and Pattern Recognition*, 2:456–463, 2001.

[2] Matthew Brand. A direct method for 3d factorization of nonrigid motion observed in 2d. *In Proc. Int. Conf. Computer Vision and Pattern Recognition*, 2:122–128, 2005.

[3] Christoph Bregler, Aaron Hertzmann, and Henning Biermann. Recovering non-rigid 3d shape from image streams. *In Proc. Int. Conf. Computer Vision and Pattern Recognition*, 2:690–696, 2000.

[4] C.G. Broyden. The convergence of a class of double-rank minimization algorithm. *Journal Inst. Math. Applic.*, 6:76–90, 1970.

[5] R. Fletcher. A new approach to variable metric algorithms. *Computer Journal*, 13:317–322, 1970.

[6] D. Goldfarb. A family of variable metric updates derived by variational means. *Mathematics of Computing*, 24:23–26, 1970.

[7] Mei Han and Takeo Kanade. Reconstruction of a scene with multiple linearly moving objects. *International Journal of Computer Vision*, 59(3):285–300, 2004.

[8] Conrad J. Poelman and Takeo Kanade. A paraperspective factorization method for shape and motion recovery. *IEEE Transactions on Pattern Analysis And Machine Intelligence*, 19(3):206–218, March 1997.

[9] D.F. Shanno. Conditioning of quasi-newton methods for function minimization. *Mathematics of Computing*, 24:647–656, 1970.

[10] Richard Szeliski and Sing Bing Kang. Recovering 3d shape and motion from image streams using non-linear least squares. Technical Report Series CRL 93/3, Cambridge Research Lab, March 1993.

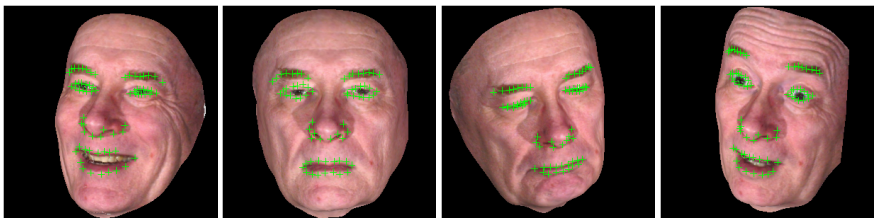
[11] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[12] L. Torresani, A. Hertzmann, and C. Bregler. Learning non-rigid 3d shape from 2d motion. *In Proc. NIPS 2003*.

[13] Jing Xiao, JinXiang Chai, and Takeo Kanade. A closed-form solution to non-rigid shape and motion recovery. *International Journal of Computer Vision*, 67(2):233–246, 2006.

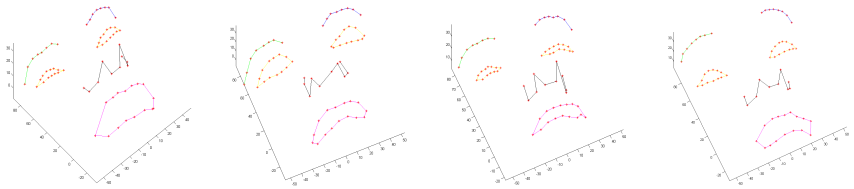
[14] Jingyu Yan and Marc Pollefeys. A factorization-based approach to articulated motion recovery. *In Proc. Int. Conf. Computer Vision and Pattern Recognition*, 2:815–821, 2005.

[15] Lijun Yin, Xiaozhou Wei, Yi Sun, Jun Wang, and Matthew Rosato. A 3d facial expression database for facial behavior research. *In Proc. 7th International Conference on Automatic Face and Gesture Recognition*, pages p211–216, 2006.



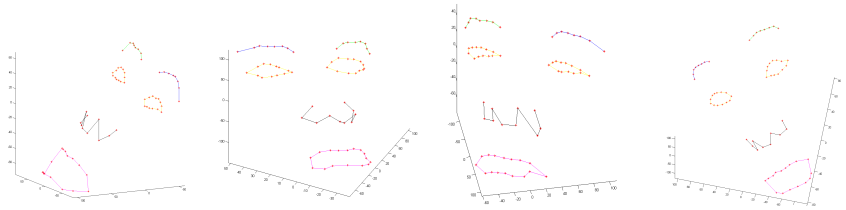
(a) **(b)** **(c)** **(d)**

Figure 2: (a) Happy expression image with rotation about $x = -10°$, $y = 20°$ and $z = 10°$ (b) Neutral expression image with rotation about $x = 0°$, $y = 0°$ and $z = 0°$ (c) Sad expression image with rotation about $x = -10°$, $y = -20°$ and $z = -10°$ (d) Surprise expression image with rotation about $x = -10°$, $y = 20°$ and $z = 10°$.



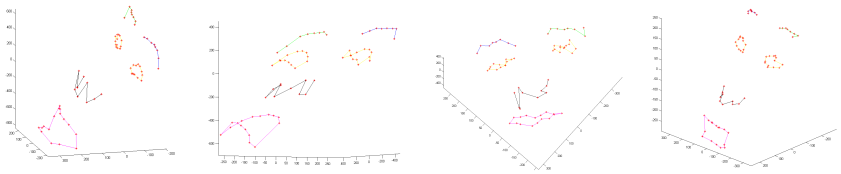
(a) **(b)** **(c)** **(d)**

Figure 3: (a) Ground truth of 3D happy expression (b) Ground truth of 3D neutral expression (c) Ground truth of 3D sad expression (d) Ground truth of 3D surprise expression.



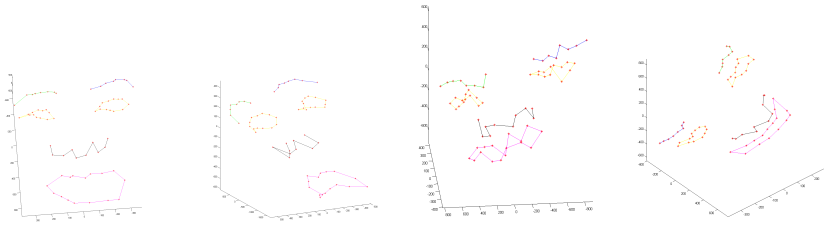
(a) **(b)** **(c)** **(d)**

Figure 4: (a) Reconstructed 3D happy expression (b) Reconstructed 3D neutral expression (c) Reconstructed 3D sad expression (d) Reconstructed 3D surprise expression under 0% Gaussian white noise.



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

Figure 5: (a) Reconstructed 3D happy expression (b) Reconstructed 3D neutral expression (c) Reconstructed 3D sad expression (d) Reconstructed 3D surprise expression under 5% Gaussian white noise.



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

Figure 6: (a) Reconstructed 3D happy expression (b) Reconstructed 3D neutral expression (c) Reconstructed 3D sad expression (d) Reconstructed 3D surprise expression under 10% Gaussian white noise.

Retinal Sampling, Feature Detection and Saccades; A Statistical Perspective.

N.A.Thacker[1] and E.C.Leek [2]. neil.thacker@manchester.ac.uk
1: Univerity of Manchester, 2: University of Bangor.

Abstract

This paper applies statistical design principles to a simple biological model of human vision so that we can more clearly interpret the apparent role of eye saccades. In doing so we show that many structural features of the biological system (such as the optical geometry of the retina) are strategies for minimising the resources required to construct a working image recognition system. The ideas presented have implications for the construction of artificial (computer) vision systems. The computational model is very closely related to (but not based upon) SIFT, but more strongly based on a consideration of vision as a process of measurment while also linking the idea of multi-scale analysis with biological structure.

Introduction

It is generally assumed that a primary function of eye movements (saccades) is to maximize information processing within the high resolution region of the fovea [19, 20]. The measurement of saccades has been widely used in psychophysical studies in domains such as visual search, reading, scene exploration and interpretation and 2-D pattern recognition [7, 3, 14], and in machine vision studies, for example, of knowledge-based scene analysis and human interactions with complex displays [4, 8] These studies have shown that eye movements can be influenced by a variety of factors, including low- level image statistics, prior knowledge and task requirements. Surprisingly, little is known about the relationship between eye movements and three-dimensional object recognition. The problem is of course complicated by the difficulty of understanding exactly what data is provided by the visual process for the task.

In this paper we examine three fundamental questions. First, can fixation data reveal preferences for specific types of image features? Second, are gaze preferences consistent between stimulus encoding and recognition? Third, are extracted features invariant across tasks and across changes in 3-D viewpoint? In order to interpret the results of this study we must first define the data we beileve to be available from the retina. The first part of this paper therefore makes an argument for a simplified interpretation of biological function.

There are several structural properties of the retina which are well known. In particular the retina is known to have spherical geometry with logarithmic sensitivity to intensity. It is also considered reasonable to assume that the input data has radially varying spatial resolution. The evidence for this comes primarily from observations of the structure of the striate cortex and secondary visual cortex [13] (cortical magnification). This would appear to be in conflict with the known structure of the retina, where although there is such a distibution of sensors, they come in two specific types (the density of illumination

sensors (rods) reduces to zero in the fovea, while the density of colour sensors (cones) are at their greatest). What we need to remember however, is that it is in principle possible to synthesise an intensity measurement from a combination of colour ones. We argue here that the requirement to construct invariant quantites for recognition, combined with the statistical nature of the data, in particular measurement stability and consideration of a quantitative understanding of the information present in the data, allows us to arrive at an hypothesis for a simplified equivalent form. Specifically a set of homogenous regional samples from an exponentially distributed set of scales. As the retina can provide only one set of image samples at any one time, building an internal representation of the world around us requires this sensor to be moved around the scene. This process is supported in the human vision system by the saccades. In this paper we provide evidence which exclude some of the less likely generators of saccadic eye movement.

Computing Invariant Quantities

In order to understand why a detector such as the retina may have evolved it is necessary to consider the fundamental problems associated with visual recognition. Much of the process of visual recognition is understood to be template matching. This is a simple idea which is difficult to make work effectively in practice. Such a process is potentially very memory intensive unless the spatio-temporal relationships are constructed in a way which eliminates unnecessary variation in the input image, such as illumination[1], rotation and scale. This is often referred to as construction of an **invariant representation**. Many researchers in computer vision would also like to build systems with invariance to perspective changes, induced by 3D sensor or object motion. Often this can be locally approximated for small rotations by affine invariance (invariance to linear skew).

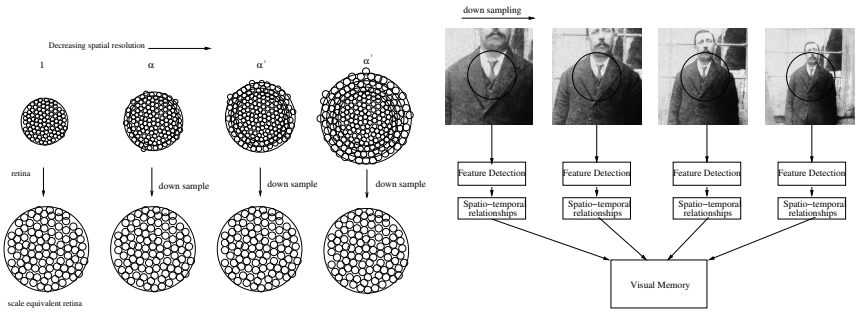
Yet combination of measured values into invariant quantities solves only part of the problem. We must also consider the associated noise characteristics. Scale and illumination invariance provide specific examples of this problem. For a CCD array the simplest way to compute illumination invariant quantities q from measured intensities $h_i \propto I + N(\sigma)$ at location i, is to compute a ratio such as $q = h_i/h_j$. Distributions of these quantities can then be treated as patterns for template matching. However, using error propagation it is possible to show that the noise characteristics of q will then be $var(q) = \sigma^2(1/h_j^2 + h_i^2/h_j^4)$. This introduces spatially varying errors on the computed invariant quantities, which then have to be properly addressed during pattern matching, by for example committing extra memory resources. The simplest way to deal with this issue is to make the invariant quantities have homogenous error by modifying the measurement process, ie: $g_i \propto log(I) + N(\sigma)$, as approximated in the human vision system. Then $q = g_i - g_j$ and $var(q) = 2\sigma^2$. One way to view this is to say, if we cannot deal with the variability introduced into invariant quantities by measurement noise then the extra information potentially gained by having a uniform sensor (h) provides no additional benefit to the alternative measurement system (g). Thus logarithmic sensitivity to light would appear to be a sensible strategy for an illumination invariant recognition system and one likely to be found in a system which has been optimised to minimise computational resources.

As with the illumination example provided above, potentially a relationship constructed from features detected at one scale can have different statistical reliability (re-

[1] I use this term here to refer to a simple linear scaing of overall intensity, and not the more general illumination variation which can occur in real scenes.

peatability error) to the same feature constructed from data at a higher scale. In the limit we can scale an object by such a factor that an object projects entirely onto one sensor. Although this is an extreme example, it is an illustration that data from a fixed sized sensor has a finite limit of information. Though the mean of any invariant quantity might remain fixed, the variance around that mean will change as a function of image scale. This will prevent reliable and efficient scale invariant recognition, which we might describe as **variable scale sensitivity**. If we wish to perform recognition with a pattern matching approach, simply constructing invariant geometric quantities is therefore not enough, as we need also to take account of the scale varying error on computed relationships.

The physical structure of the eye might appear unnecessarily complicated in comparison to a simple colour CCD array. To begin with the retina is a curved (approximately hemi-spherical) surface whereas a CCD array is flat. The optical model for a conventional electronic device is close to a pin-hole model. The geometric imaging process is described as perspective projection. Under such a model objects viewed at the centre of the field of view appear different to those at the edge. A spherical imaging surface on the other hand is rotation invariant and will produce an equivalent focused image of an object for any position in the field of view. This can be considered a simple form of perspective invariance. However, such a property pre-supposes a uniform sampling of the image, which is manifestly not the case for the human vision system.



(a) (b)

Figure 1: Computing scale invariant samples in the retina (a) and Scale invariant recognition (b).

One way to avoid the problem of variable scale sensitivity is by designing our image sensor so that it delivers a scale invariant measurement. We don't need to believe that this is exactly the biological solution at this stage, but if we cannot produce a scale invariant measurement system then we certainly cannot expect to produce any other scale invariant quantities[2]. As it is easier to consider the issue of scale invariance in cartesian geometry, we start on a planar sensor in what we will call the fovea where we assume the data has approximately homogenous spatial sampling [3]. We wish to synthesise a new sensor now at a lower resolution ($1/\alpha$). This process is illustrated in Figure 1a. The inner values of this new sensor can be computed from the fovea using a process of down-sampling. The most likely computational form for this process is a Gaussian convolution (or its irregular

[2]We are not talking here about the approximate invariants generally used in computer vision, but fully invariant quantities obtained by constructing a sensor so that such things are computable.

[3]As discussed above, in the biological system this process is complicated by the presence of two distinct types of light detector, rods and cones.

sampled equivalent) as this is the only scale invariant sub-sampling process, due to the central limit theorem. The outer values of the new region are unavailable in the fovea and therefore require additional sensors around the edge of the previous active region.

The new ring of sensors need only to have a spatial resolution which is a factor α larger than the ring of original samples on the edge of the fovea. To have better spatial resolution is potentially wasteful with resources. Down sampling once more, in order to construct the next equivalent sensor, we can again subsample the inner regions and we will need a new layer of measurements from areas now α^2 larger than the original. As this process continues we gradually construct a sensor surface, with ever decreasing spatial resolution. The mathematical process we have just described for positioning each ring of new sensor locations is

$$r_n = r_0(1 + k \sum_{i}^{n} \alpha^n)$$

Outside of the fovea, the radial function which characterises this process is obtained if the spacing of sensors is exponential outside of the fovea, (ie: a logarithmic sensor). This model differs from previous interpretations [13] [4], in that it accomodates the uniform sample density in the fovea as an intrinsic part of the calculation of scale invariant shape represenatation.

Making the adjustment from a planar sensor to another geometry, such as locations as angles ϕ on a spherical sensor, is mainly a case of applying the appropriate transform [5].

In conclusion, the equivalent homogenous sampling interpretation for feature detection on the retina now allows us to exploit the sampling properties of a spherical optical geometry. In particular, the spatial distribution of uniform sensors in the fovea can be used to compute the locations that sensors need to be placed when the same object is viewed at a larger scale in order to recieve identical input data. Such a scheme provides a partial invariance to projective deformation which is absent in a conventional electronic device.

An Overview of Statistically Based Feature Detection

We would like to take methods from computer vision as candidates for feature extraction in the brain. The topic of computer vision represents a large body of literature and in most cases the difference in behaviour and capabilities of the methods are heavily influenced by intended use. There are consequently as many feature detectors as there are applications. However, the general principles involved for much of this work can be characterised as template matching and interest operators. Both of these are physiologically viable methods for scene analysis. Template matching is analogous to the concept of receptive field patterns, while interest operators are more akin to the hypothesised mechanisms related to the possible role of micro saccades. The following section makes an argument for interpretation of these approaches as different aspects of the same underlying principle for information extraction.

Scaled features can be computed at a range of spatial scales by processing regular equivalent sensors with a fixed set of feature detectors, or (more likely) by combining

[4] Where an adjustment to the logratihmic form $log(r) \rightarrow log(r+a)$ is made which breaks the pure scale invariance property exactly where we might expect it is most needed, in the fovea.

[5] Actually, this argument to apply exactly we need the initial definition of the sample on the fovea to be equivalent to an homogenous sampling on a spherical surface not a planar one. The difference here is negligible for a fovea of small angular extent.

sub-sampling with the process of feature detection. The equivalent sample images need never be explicitly computed, but the resulting output of the system is equivalent to if they had been. Scale invariance can then be achieved by applying a selection process to choose the one scale most suitable for representation of the local image patch (Figure 1b). The selected representation at a single scale is then compared to stored visual memory. If the viewed object changes scale, by for example moving towards the viewer, then the scale at which selected features are detected will change so that the same, scale invariant, spatio-temporal relationships are selected for pattern matching. This process is effectively what is advocated in the Scale Invariant Feature Transform (SIFT) [10].

The best known of the template based approaches include the Canny edge detector [2] and the Difference of Gaussian operator [12]. These methods apply combinations of image convolution operations in order to enhance selected features. Connected or isolated feature points can then be identified as maxima on these enhancement images, in this case step edges and ridge locations respectively. The main difference between these two approaches is that although a step edge detector will respond to all but the finest of ridge features, Difference of Gaussian processing will not enhance step edges [6] (the most common manifestation of an object boundary) and has an output which is more strongly dependant upon feature scale and illumination.

In order to take advantage of the inherent properties of illumination invariance, feature detection processes need to be computed as combinations of differences between measured values g. For many features defined in computer vision, such as conventional first derivative based edge detectors, this is clearly the case.

$$e \;=\; G(\sigma) \otimes \sqrt{(g_{i+1} - g_{i-1})^2 \;+\; (g_{j+1} - g_{j-1})^2}$$

where $G\otimes$ represents a Gaussian convolution of width σ. Interestingly e^2 is a smoothed local estimate of the Fisher Information associated with local image plane orientation.

Although using a multitude of template feature detectors, all matched to distinct feature types, is a possible algorithmic solution for the extraction of object structure, this approach raises an important question; What is a valid mechanism to combine the responses from the different detectors? This must be done in a way which provides generalisation for recognition of shape, not only for changes in illumination and object scale, but also over possible responses to changes in scene content (for example arbitrary possible backgrounds at object boundaries). Interest operators provide an alternative which is capable of detecting many characteristic feature types with one simple computation. The simplest interest operator would be a local variance estimate of image signal, which when applied at a range of spatial scales, is also useful as a descriptor of texture [5]. For example;

$$v \;=\; G(\sigma) \otimes (g - < g >)^2 \quad \textit{where} \quad < g > \;=\; G(\sigma) \otimes g$$

which can be considered as either the etimate of signal to noise associated with local image variation, or the inverse of Fisher Information associated with a mean value. This simplifies to;

$$v \;=\; G(\sigma) \otimes g^2 \;-\; (G(\sigma) \otimes g)^2$$

[6] The response to a step edge from a DoG filter (as advocated in SIFT), is exactly zero at the position of the edge. Although it takes large positive and negative values on either side, the locations of the maxima are systematicaly shifted as a function of the Gaussian kernel size and therefore cannot be considered as spatially consistent.

Notice that these calculations embed directly Gaussian convolutions, which we have already stated are required for scale invariant resampling.

Another popular feature detector in computer vision is the corner detector and one approach uses the concept of an interest operator which is often based on the idea of auto-correlation. Corner detection can also be performed with templates, but is significantly more difficult than edge detection due to additional variation in orientation and corner shape [7]. The Harris corner detector [6] defines corner locations using the second order spatial variation of an auto-correlation around a point. However, auto-correlation can be interpreted as a log-likelihood for the degree of match between the original local image patch and a shifted version. In adddition the matrix of second order behaviour is the second term in a Taylor expansion, so it can also be interpretted as the second derivative with respect to image location. As the Cramer-Rao bound is the second derivative of a log likelihood, this is the Fisher Information for spatial localisation.

In summary we now have three definitions of feature detector based upon quantitative measurements which define positions of maximum information; local variance for spatially varying intensity, edge strength for orientation, and interest operators for spatial location. In fact, if we consider feature detection as a template based approach supported in the biology by receptive fields, then grey level scale, orientation and location are the only measurable quantities possible. It makes sense to suggest that if restrictions on processing capacity (for example finite connectivity) results in the need for the brain to identify a subset of features in order to solve quantitative tasks, then those which make the largest quantitative contribution (ie. those which maximise some aspect of Fisher information) are the ones it should be using and the ones we should be basing any model of scene interpretation upon.

Finally, illumination invariance of these measures and also for colour is entirely reliant upon logarithmic sensitivity to light. As with spherical optical geometry, this is a property which is lacking in a conventional electronic sensor. It is becoming increasingly obvious that when it comes to getting a simple solution to visual analysis tasks, the biological sensor has characteristics which make a lot of sense. Indeed, the analysis of data from a conventional colour CCD array will be difficult by comparison.

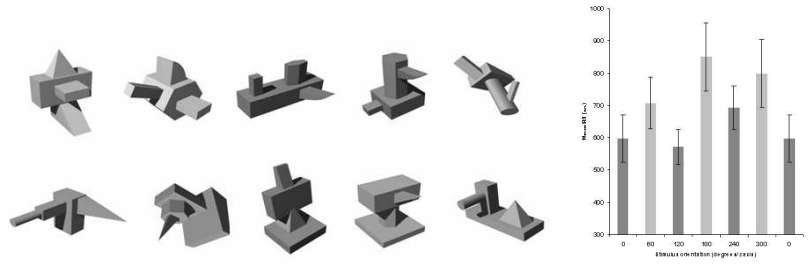
Assuming that the process of shape recognition is based upon conjunctions of detected features, then the above description of a multi-scale feature detection process eliminates the need for scale invariance. If we also eliminate the possibility of full 3D rotation invariance (on grounds of in-homogenous error characteristics), the required invariances for a shape representation are therefore translation and rotation within the sensor "plane". An ideal representation of shape would be one which supported the reconstruction of the shape up to an unknown position and orientation. Such a representation has been previously described as "complete". Although simple regional histograms of local image orinetation (as used in SIFT) are not complete [16], the property was established over a decade before for the representation scheme referred to as "pairwise geometric histogrames" [15]. This approach provides an encoding of local shape as a 2D frequency distribution of relative angle against perpendiular distance.

Methods: Investigating Patterns of Eye Movement

The simplest hypothesis for the role of saccades is that we move our eyes in order to build up a high resolution measurement of the scene. This hypothesis can be immediately ex-

cluded by observing real eye movements, which do not uniformly scan the potential view field, but seem instead to be drawn to particular visual features, movement or objects. A more sophisticated hypothesis for the role of saccades in visual exploration is that we saccade to areas which are expected to have useful spatial information for the interpretation of shape or structure [19]. We would therefore expect the eye to saccade to those features which are most useful for this task. As we have only low resolution data available in the periphery of the retina, we must assume that this is somehow used to predict the most useful places for fixation. Although we may not know precisely what the human vision system does, we suggest here that we can take the standard feature detection processes as characterised by interest operators and template matching approaches as indicative of those features which would be useful for the purpose of extracting image structure. We can then see to what extent the saccadic process targets locations in images which contain structural features in order to examine our initial hypothesis.

In brief, participants (N = 24; Mean age = 22.67) first viewed sets of six novel 3-D objects each containing one principal component and three sub- components or volumetric parts (see Fig 3a). 12 objects (6 targets and 6 distractors) were presented from three different viewpoints (0, 120, 240 degrees) each for 10 seconds while eye movement patterns were recorded. Following the Learning Phase, participants performed a recognition memory task in which they had to discriminate learned from unfamiliar objects, presented either at practiced (0, 120, 240 degrees) or novel orientations (60, 180, 300 degrees) in depth. Behavioural responses (accuracy and Reaction Times (RT)) were recorded. Eye movement data were recorded on a Tobii ET17 remote eye tracking system running at a data acquisition rate of 50 Hz. Experimental stimuli were viewed from a distance of 60 cm at a screen resolution of 1280 x 1024.



(a) (b)

Figure 2: Novel objects (a) and reaction times for recognition (b).

Results

Accuracy of target detection in the Test Phase was very high (range 80- 94.17 %). As expected, targets were detected more accurately at the practiced viewpoints, F (1, 23) = 26.01, $p < .001$. Mean RTs for correct trials are shown in Fig 3(b). A 2 (Familiar vs Familiarity) x 3 (Viewpoint) repeated measures ANOVA showed that RTs were faster for practiced over unfamiliar viewpoints, F (1, 23) = 13.73, $p < .001$. The main effect of Viewpoint was not significant. There was no interaction.

Analyses of eye movements were conducted by initially pre-processing raw gaze data by applying spatial and temporal filters to remove micro-saccades and drift. Fixations were defined as eye movements that remain within the same circular region (diameter 60

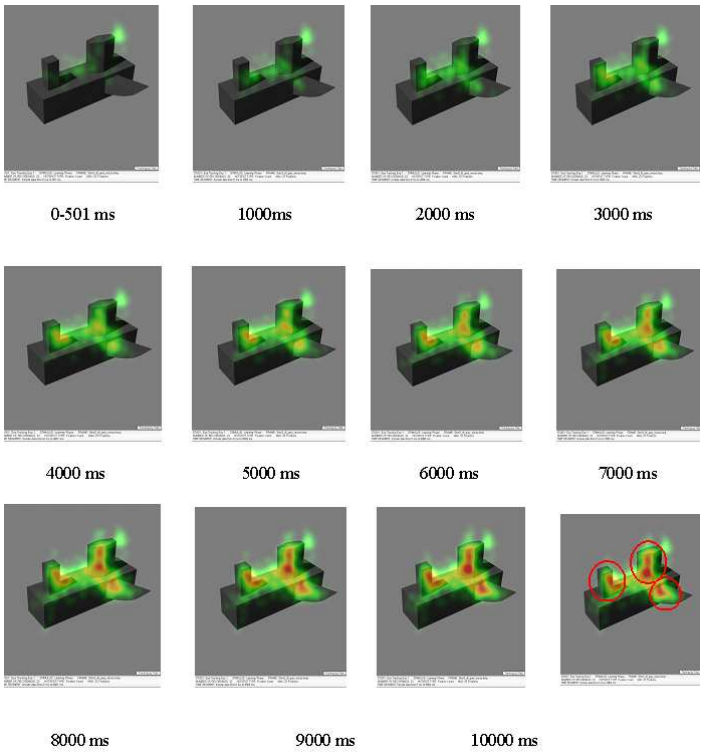
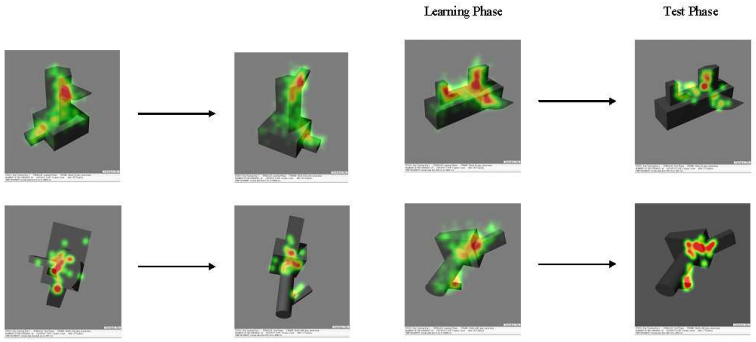


Figure 3: Time development of fixations.

px) for a minimum of 100 msecs. Filtered data were used to compute fixation frequency across participants for each stimulus. Figure 4 shows a time series fixation frequency plot for all participants overlaid onto the original stimulus image. The data are grouped into 10 epochs corresponding to the first 500 msecs post stimulus onset, and then for each 1000 msecs thereafter. The data show that participants rapidly fixate on image regions which appear to correspond to salient 3-D image segment points around object sub- components. Figure 5(a) shows an analysis of the consistency of fixations across changes in the 3-D viewpoint for two of the test stimuli. This shows that participants consistently search for and fixate the same 3-D image segmentation points across viewpoint, despite changes in the low-level image properties of these locations (e.g., vertex types). Figure 5(b) shows an analysis of the consistency of fixations between the Learning and Test Phases for two items. As in the previous analysis, participants fixate the same image regions between phases.

Conclusions

This paper has sought to explain saccadic eye movement within a framework which includes some of the more obvious structural features of the human vision system. It seems to be possible to account for many observed properties, including logarithmic intensity sensitivity, and spherical optical geometry, in terms of construction of invariant repre-



(a) (b)

Figure 4: Changes across viewpoint (a) and consistency between learning and test phases (b).

sentations which take account of measurement error. The kinds of algorithms generally developed in the area of computer vision, with regular input lattices and at fixed scales, may seem a world away from irregular sampling of the retina and saccades. However, it seems possible to replicate the process of scale analysis by simply processing at multiple scales and selecting one result. This opens the possibility of applying insights from image analysis to interpretation of visual biological.

Ultimately the brain must use detected features for the construction of shape representations. The brain will need to analyse the incoming spatio-temporal data to extract compact descriptions for the purpose of accurate prediction and categorisation. Analysis of the statistical nature of the data tells us that it is not possible to construct a representation which is invariant to every form of variation produced during image formation. However, invariances are key to the construction of efficient vision systems, as the more we can correctly generalise from data we have already learned and understood, the easier it is to interact with our environment. The development of invariant recognition processes could be invoked as an implicit target during the process of human evolution, thereby explaining the kind of structure we see on the retina today.

Our study supports the following conclusions: (1) Human visual biology is consistent with a simple structural hypothesis (based on multi-scale samples) for the construction of invariant recognition systems which opens the way for interpretation of retinal data using conventional machine vision approaches, (2) Fixational eye movement patterns during 3-D object recognition are not random, but rather structured and highly consistent among observers. (3) There is remarkable consistency in the patterns of fixational eye movements shown for 3-D novel objects across both changes in viewpoint and between learning and test phases. (4) These locations show evidence of 'top down' selection and are **not** the low level features generally constructed for machine vision.

These conclusions are not dependent upon the particularly simple nature of our stimuli, and tell us that tracked features are a long way from the input visual data, in terms of processing. They imply a high level representation of 3D structure which is already available prior to eye movement. The most striking observation is that fixated locations are often on surfaces, which in our data at least contained no low level information. At first sight this may seem to be at odds with a feature based analysis of shape. However,

these conclusions can be reconciled with a feature based analysis if we take a view based approach to recognition, whereby sets of features within a focal region are used for shape representation such as a learned set of geometric co-occurences (such as the PGH). We can have every reason to believe that fundamental understanding of the problems involved in extracting shape information from conventional images is potentially of direct relevance to understanding high level processing in human vision. This being the case, locations of saccadic fixation should contain valuable information which can help identify these high level processes. This is an avenue we intend to explore further.

References

[1] A.P.Ashbrook, N.A.Thacker, P.I.Rockett and C.I.Brown, 'Robust Recognition of Scaled Shapes Using Pairwise Geometric Histograms.', proc., BMVC 95 Birmingham, 503-512, July 1995.

[2] J. Canny., A computational approach to edge detection., IEEE Transactions on Pattern analysis and Machine Intelligence, 8(6),679-698, 1986.

[3] A.T. Duchowski, Eye Tracking Methodology: Theory and Practice. Springer. London, 2003.

[4] J.M. Findlay and I.D. Gilchrist, Eye guidance and visual search. In G. Underwood (Ed.), Eye Guidance in Reading and Scene Perception. Oxford. Elsevier, 1988.

[5] R.M. Haralick, Statistical Image Texture Analysis, Handbook of Pattern Recognition and Image Processing, Ed. T.Y Young and K.S. Fu, Academic Press, Orlando, 247-279, 1986.

[6] C.Harris and M.Stephens., "A Combined Corner and Edge Detector" Proceedings of the Fourth Alvey Vision Conference. 147-151, August 1988.

[7] J.M. Henderson, C.C. Williams, M.S. Castelhano, R.J. Falk, Eye movements and picture processing during recognition. Perception and Psychophysics, 65, 725-734, 2003.

[8] R.S. Johansson, G. Westling, A. Backstrom, J.R. Flanagan, Eye-hand coordination in object manipulation. Journal of Neuroscience, 21, 6917-6932, 2001.

[9] A.J.Lacey, N.A.Thacker and N.L.Seed. 'Smart Feature Detection Using an Invariance Network Architecture', proc., BMVC 95 Birmingham, 327-336, July 1995.

[10] D.G.Lowe, Distinctive Image features from Scale-Invariant Key-points, Int. Jou. Comp. Vis, 2004.

[11] S. Martinez-Conde, S.L. Macknik and D. H. Hubel,, The Role of Fixational Eye Movements in Visual Perception, Nature Reviews Neuroscience, 5(3), 229-238, March 2004.

[12] T. Peli and D. Malah., A study of edge detection algorithms., Computer Graphics and Image Processing, 20, 1-21, 1982.

[13] E.L.Schwartz, Spatial Mapping in the Primate Sensory Projection: Analytic Structure and Relevanve to Perception, Biol. Cyber., 25, 181-194, 1977.

[14] K. Schill, E. Umkehrer, S. Beinlich, G. Krieger, C. Zetzsche, Knowledge-based scene analysis with saccadic eye movements Human vision and electronic imaging IV; Proceedings of the Conference, San Jose, 520- 532. 1999.

[15] N.A.Thacker, P.A.Riocreux, and R.B.Yates, 'Assessing the Completeness Properties of Pairwise Geometric Histograms", Image and Vision Computing, 13, 5, 423-429, 1995.

[16] N.A.Thacker and J.E.W.Mayhew, 'Designing a Network for Context Sensitive Pattern Classification.' Neural Networks 3,3, 291-299, 1990.

[17] N.A.Thacker, I.A.Abraham and P.Courtney, 'Supervised Learning Extensions to the CLAM Network.' Neural Networks Journal, 10, 2, pp.315-326, 1997.

[18] N.A.Thacker, F.Ahearne and P.I.Rockett, 'The Bhattacharyya Metric as an Absolute Similarity Measure for Frequency Coded Data.' Kybernetika, 34, 4, 363-368, 1997.

[19] L. Walker and J. Malik. Sequential information maximisation can explain eye movements in an object learning task. Journal of Vision, 4, 744a. 2004.

[20] A.L. Yarbus, Eye Movements and Vision. New York. Plenum Press, 1967.

Unsupervised Category Discovery in Images Using Sparse Neural Coding

Stephen Waydo and Christof Koch
Control & Dynamical Systems and Computation & Neural Systems
California Institute of Technology
Pasadena, CA 91125, USA
waydo@cds.caltech.edu

Abstract

We present an unsupervised method for learning and recognizing object categories from unlabeled images. Motivated by the existence of highly selective, sparsely firing cells observed in the human medial temporal lobe (MTL), we apply a sparse generative model to the outputs of a biologically faithful model of the primate ventral visual system. In our model, a network of nonlinear neurons learns a sparse representation of its inputs through an unsupervised expectation-maximization process. In recognition, this model is used in a maximum-likelihood manner to classify unseen images, and we find units emerging from learning that respond selectively to specific image categories. A significant advantage of this approach is that there is no need to specify the number of categories present in the training set. We present classification accuracy using three different evaluation metrics.

1 Introduction

Highly sparse representations of objects in the visual environment in which individual neurons display a strong selectivity for only one or a few stimuli (such as familiar individuals or landmark buildings) out of perhaps 100 presented to a test subject have been observed in the human medial temporal lobe (MTL), a brain area crucial to the formation of new memories [11, 18]. While highly selective for a particular object or category, these cells are remarkably insensitive to different presentations (i.e. different poses and views) of their preferred stimulus. By contrast, neurons in the inferotemporal cortex (IT), immediately earlier in the visual pathway, respond in a much less sparse manner [14]. A natural question to ask is thus "how do neurons in the MTL learn their sparse and invariant representations from the incoming visual information?" From a machine vision standpoint, this question can be viewed as a problem in unsupervised image classification: given a set of unlabeled training images, can we design an algorithm that will group these images into categories corresponding to those human observers would impose? This is clearly distinct from the more common approach to object recognition in which a labeled training set is used to learn features common to the category which can then be used to classify unlabeled images [1, 4].

Motivated by the neurobiological results, we study the effects of applying a sparse-coding model to the outputs of a biologically faithful model of the primate ventral visual

cortex [13, 15]. The sparse-coding model, which itself employs biologically plausible learning operations, is derived from that of Olshausen & Field [10], which they used to develop a sparse representation of natural images much like that observed in primate visual cortex. We seek to use a similar learning algorithm to build a representation in which individual units of our output layer respond in a selective and invariant manner to specific object categories.

1.1 Related Work

Unsupervised image classification has only recently begun to attract attention in the literature. Sivic et al. [17] apply techniques from unsupervised topic discovery in text to "words" derived from SIFT descriptors to discover categories in images. While their approach is very different from that taken here, the problem they attempt to solve is the same (and we evaluate our results on many of the same datasets). An important distinction is that they found it important to restrict the number of categories searched for to the number truly present in their datasets, while our method is robust to varying numbers of input categories. Fergus, Perona, and Zisserman [4] use an unsupervised generative learning algorithm to build representations of particular image categories, but only images from a single category are presented to the model, which is then tested in a category-versus-background setting. In contrast, our model simultaneously learns representations for multiple image categories without *a priori* specification of the labels (or even the number of categories present). Weber, Welling, and Perona [19] also cast the unsupervised categorization problem as emergent population coding, but without the sparseness constraint that is key to our results. Serre, Wolf, and Poggio [16] developed the underlying vision system model we use here, and they show that the features generated are sufficient to classify our input categories with high accuracy (using a supervised classifier).

Sparse coding as a computational tool has attracted a great deal of attention in recent years, both in the context of vision and elsewhere. Olshausen and Field developed the algorithm we apply here and showed that, when applied to natural image patches, it generates a code much like that observed in simple cells in primary visual cortex [9, 10]. Hinton and Ghahramani [6] also cast sparse representation in a generative modeling framework, but as with Olshausen and Field they work directly at the image level. Sparse coding is closely related to *Independent Components Analysis* [2], which has been used to generate natural image codes similar to those obtained from sparse coding [3]. Li et al. [7] discuss the use of sparse representation for blind source separation, including the notion that the number of sources (in our nomenclature, categories) need not be specified, but they do not address the application we present here. Mutch and Lowe [8] improve the performance of the underlying vision system model we use here, in part using sparsification to enhance selectivity. Ranzato et al. [12] take an energy-based approach to the unsupervised learning of sparse representations of natural images and briefly discuss its extension to a hierarchical model. Both of these efforts are at a much lower level of the hierarchy and so do not address categorization.

2 Approach

We first generate an invariant feature-based representation of our images (analogous to that found in IT) using the hierarchical feedforward model of object recognition described

by Serre et al. [15] and available at `http://cbcl.mit.edu`. The output of this stage - applied to many images from several different image categories - is then sent into a sparse coding model (modified from [10]). This network attempts to identify sparse structure in its inputs via unsupervised learning on sample input data. To evaluate performance we examine the selectivity of the trained network to unseen images from the same categories as the training data.

2.1 Input Processing

All images used in this investigation were taken from the Caltech-256 database of images from 256 categories [5]. Images were resized (using MATLAB's `imresize` with nearest-neighbor interpolation) so that the smaller dimension was 128 pixels while preserving the aspect ratio. The outputs of the C2b and C3 layers of the visual processing model [15] were computed using a feature set derived from training on 500 natural images (no new features were learned - this investigation used the filters included in the standard distribution of this model). There were 1000 units in each of these layers, for a total of 2000 outputs. These outputs were then normalized so that each output unit's responses had zero mean and unit variance across the input set for a given experiment. These normalized outputs were used as inputs to the sparse coding model described below.

2.2 Sparse Coding

We seek to build a generative model $\mathscr{G}$ of the inputs $u \in \mathbb{R}^n$ (here, $n = 2000$) to our model with the assumption that there exists some sparse set of causes $v \in \mathbb{R}^m$ (with $m \ll n$) underlying the observed data. In our case the u are the responses of the underlying vision system model to the input images, while each element v_i of v will come to represent an image category. In general, we wish to find probability density functions $f(v|\mathscr{G})$ and $f(u|v,\mathscr{G})$ such that the distribution of generated inputs

$$f(u|\mathscr{G}) = \int_v f(u|v,\mathscr{G}) f(v|\mathscr{G}) \tag{1}$$

closely matches $f(u)$, the distribution of inputs observed in the training data. Once such distributions have been found, we can attribute causes to inputs by a deterministic maximum-likelihood process, or

$$v(u) = \arg\max_v f(v|u,\mathscr{G}). \tag{2}$$

Following the approach of Olshausen & Field [10], we can use this framework to search for a sparse code for our inputs. First, we assume the causes underlying the inputs are sparse and independent, setting

$$f(v|\mathscr{G}) \propto \prod_{i=1}^{m} \exp(S(v_i)), \tag{3}$$

where $v_i \in \mathbb{R}$ is the i^{th} element of v and $S(v_i)$ is defined such that the resulting distribution is sparse. For simplicity we omit the proportionality constant required to make this distribution integrate to 1. In [10], where this strategy was used to develop a V1-like sparse code for natural images, the sparse prior S followed a Cauchy distribution. Because we

seek to develop units that respond in a more-or-less binary fashion (i.e. most responses are close to 0 or 1), we instead use a weighted sum of two Gaussians with variance σ^2, one centered at 0 with weight $1-t$ and the other at 1 with weight t.

Second, we assume that the distribution of inputs given a cause is Gaussian with a mean given by a linear function of the causes, that is $E[u] = Gv$ for some $G \in \mathbb{R}^{n\times m}$, and diagonal covariance matrix $COV[u] = \lambda I$. The columns of G are thus basis functions for representing the inputs u. We further place a zero-mean Gaussian prior distribution with variance γ^2 on the elements g_{ij} of G to avoid an extra normalization step required in earlier work.

Our generative model $\mathscr{G}$ is now parameterized by the matrix G. The function to be maximized with respect to G is the average log-likelihood of the data within the model,

$$\begin{aligned}\mathscr{F}(v(u),G) &= \langle \ln f(v(u),u,G)\rangle \\ &= \left\langle -\frac{1}{2\lambda}\|u-Gv(u)\|^2 + \sum_{i=1}^{m} S(v_i(u)) - \frac{1}{2\gamma}\sum_{i=1}^{n}\sum_{j=1}^{m} g_{ij}^2 \right\rangle.\end{aligned}$$

The first term in $\mathscr{F}$ penalizes a mismatch between the true input u and the modeled input $Gv(u)$, the second term rewards responses that are likely according to the sparse prior, and the third term penalizes large weights in G.

We optimize this function via expectation maximization. In the E phase, for each input u we seek to compute the most likely cause $v(u)$ (i.e. the arg max of $\mathscr{F}$). Performing gradient ascent on $\mathscr{F}$ with respect to v we obtain the differential equation

$$\dot{v} = \frac{1}{\lambda}G^T(u-Gv) + S'(v) \tag{4}$$

where the vector-valued function $S(v)$ is shorthand for S evaluated on each v_i and S' is the derivative of S with respect to v. This system can be implemented as a two-layer recurrent neural network with nonlinear dynamics in the output layer given by S'. This stage of the optimization computes the set of basis functions that best represent the input, subject to the sparseness constraint imposed by S.

In the M phase, we compute the optimal G for the current $v(u)$. Taking the derivative of $\mathscr{F}$ with respect to G, setting equal to zero, and solving for G we obtain the update rule

$$G \rightarrow \langle uv^T\rangle \left(\frac{\lambda}{\gamma}I + \langle vv^T\rangle\right)^{-1}. \tag{5}$$

This rule yields the global optimum for G given $v(u)$ and so lets us take large steps toward the optimum of $\mathscr{F}$ in the M phase. This in turn leads to much faster convergence than the incremental update used in previous work [9, 10]. If, however, we wish to perform on-line learning in which images are presented one at a time, gradient ascent yields a Hebbian-with-decay update rule.

3 Classification Experiments

We performed several experiments with this model. In all cases the number of outputs from the C2b and C3 layers of the visual system model [15] - and thus the input to the

sparse learning network - was $n = 2000$, and the number of output units was $m = 10$. The matrix G was initialized with uniformly distributed random weights between -0.5 and 0.5. Equation 4 was simulated in MATLAB for sufficient time to reach equilibrium with the additional constraint that all responses v_i be nonnegative (using MATLAB's "NonNegative" `odeset` property) and parameters $\lambda = 10$, $t = 0.05$, and $\sigma^2 = 0.04$. The weight penalty was $\gamma = 100$. In each experiment we used the batch update rule (eq. 5) and terminated the optimization when the average change in the weights g_{ij} was less than 1%. Except for experiment (D), for which fewer images were available, we used 40 random images from each category for training and reserved 40 different images for testing. After training, the recognition model (eq. 4) was run on the novel testing images.

We performed the following four experiments:

(A) Three object categories. The model was trained and tested on images of motorbikes, airplanes, and faces. This is directly comparable to experiment (C) of [17].

(B) Four object categories. We added a fourth category (cars) to the training set from experiment (A). This is similar to experiment (D) of [17], except that we used side- rather than rear-views of cars.

(C) Four object categories. As the images from experiment (B) are relatively easy to classify (a supervised classifier operating on the same inputs can perform this task at near 100% accuracy), we performed the same experiment with four more difficult categories: blimps, elephants, ketches (a type of sailboat), and leopards.

(D) Five individuals. We sorted the face images from the Caltech 256 database into categories consisting of images of the same individual. We then presented images of 5 of these individuals. We presented 10 images of each individual in the training stage, reserving 10 different images of each individual for testing.

4 Results

We ran each experiment 10 times with different random initial conditions for G. All model parameters were identical between the four experiments - no adjustment was required to account for different number or type of input categories between experiments.

4.1 Response Profiles

We here focus on describing the response profiles of the output units from a typical run of experiment (B); results from the other trials and experiments were qualitatively similar. Figure 1 depicts the responses of two of the selective units (from the same session) that emerged in training. For each unit this figure shows 20 of the 40 images that evoked the strongest responses (every other response is omitted for clarity) as well as a histogram of all responses. The ROC curve for each unit treated as a classifier for its preferred category is inset in the histogram, along with the ROC curve for the best principal component for that category for comparison. We see from these figures that category tuning has spontaneously emerged from the learning process.

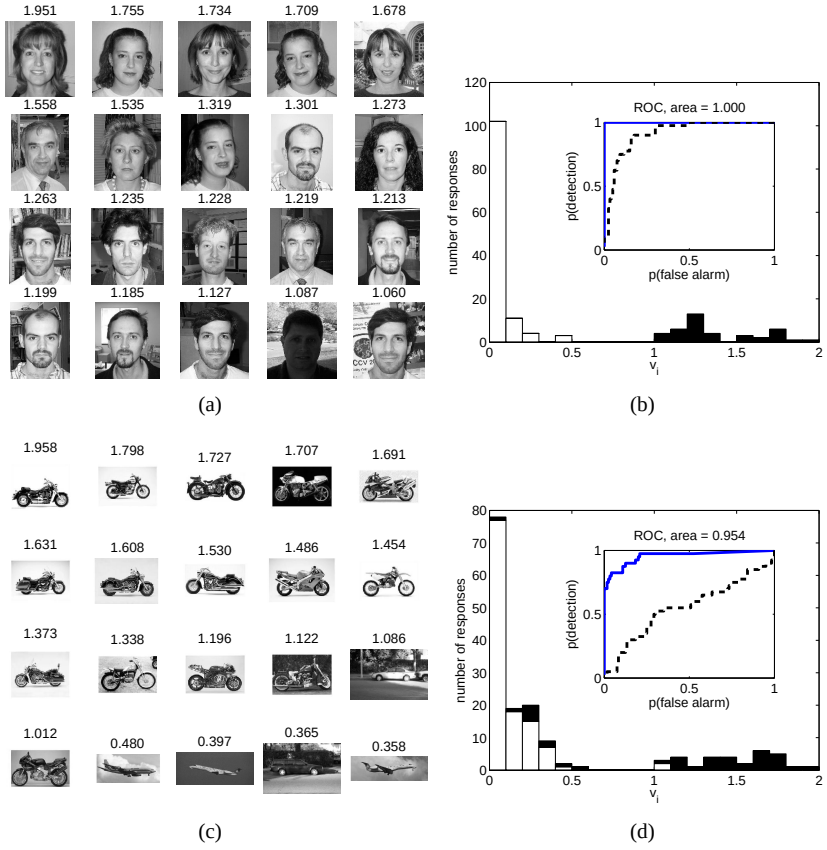


Figure 1: Responses of two selective units after the unsupervised category learning. (a,c): images that evoked the top responses, with the activation level above each image. Every 2^{nd} image omitted for clarity. (b,d): response histograms. *x*-axis is the activation level; *y*-axis is the number of test images (160 total) evoking a response at that level. Responses to preferred category in black; responses to all other images in white. Insets: ROC curves. Solid line is ROC curve for selected unit, dashed line is ROC curve for best principal component. ROC equal-error accuracies were 100% and 88%.

4.2 Classification Accuracy

Given that we use a purely unsupervised training process, and that our model is free to identify fewer or more categories than are present in the training set, there are several possibilities for evaluating the classification accuracy of this system. We consider three metrics here, two of which are weakly supervised as they require us to decide what category each unit is selective for, and one of which is fully unsupervised:

Metric 1: Single-category classifier. We consider each unit individually as a classifier for its most selective category. The accuracy figure we use is the receiver-operating characteristic (ROC) equal error rate (i.e. p(true positive) = 1-p(false positive)) testing against the other categories. Chance level in this case is 50%. The metric is the average accuracy of our best classifier for each category.

Metric 2: Weakly supervised classifier. We use all selective units together to classify each input image into one of the input categories. To do so, we first manually assign to each unit a category for which it is most selective as before (so multiple units could be assigned the same category). We then classify each image according to which unit responded the most strongly. The accuracy is then the percentage of testing images correctly classified, and the chance level is one over the number of categories.

Metric 3: Unsupervised classifier. In the fully unsupervised setting we rely on the output units to both define the categories and assign images to them. Each image is assigned to a putative category based on which output unit responded the most strongly. We then form a confusion matrix in which element (i, j) is the percentage of images from input category j assigned to output category i and rearrange this matrix to maximize the average of the diagonal elements, thereby picking the output categories that best correspond to the input categories. This average is then the classification accuracy, and chance level is one over the number of output units (in this case 10).

Note that each of these metrics says something different about the behavior of the network, and none of them by itself describes exactly the sparse, invariant selectivity that is our goal. Metric 1 quantifies how selective individual units are for particular categories, but disregards the separation between on- and off- responses. Metric 3 quantifies how precisely the categories discovered by the network correspond to those we defined, but a network that divides one or more categories into subcategories would score poorly here despite qualitatively good performance. Metric 2 alleviates this issue, but could disregard excessive subcategorization. Hence, sparse, invariant representation of the input categories is only captured by good scores according to all three metrics.

The results of each experiment as measured by these metrics averaged over 10 trials are summarized in Table 1. As a baseline for comparison, we also evaluated the performance of Principal Components Analysis (PCA) applied to the same inputs as our sparse coding network against these three metrics. As we had 10 units in the output layer of the sparse coding network, we used the top 10 principal components for this comparison. We also found the best performance we could achieve using a supervised SVM classifier applied to the same inputs, which provides a reasonable upper bound on achievable performance and an objective measure of task difficulty. For metric 1 we report the average accuracy of a binary SVM classifier for each category versus the others, while for metric

Ex	Metric 1				Metric 2			Metric 3			
	SN	PCA	SVM	ch	SN	PCA	ch	SN	PCA	SVM	ch
A	91.7	69.2	98.1	50.0	90.6	55.0	33.3	64.0	37.5	96.7	10.0
B	89.8	71.9	97.4	50.0	82.6	46.9	25.0	66.1	40.6	96.9	10.0
C	77.0	69.2	88.1	50.0	63.8	47.5	25.0	41.4	36.3	81.9	10.0
D	94.8	85.0	98.0	50.0	83.6	62.0	20.0	75.0	70.0	100.0	10.0

Table 1: Classification accuracy computed using different metrics averaged over 10 trials with random initial conditions. In all cases unseen images were used for testing. For each metric we report the classification accuracy (as a percentage) for the sparse network (SN) and for PCA applied to the same inputs, as well as chance level. For metrics 1 and 3 we also provide the accuracy of a supervised SVM classifier applied to the same inputs.

3 we report the accuracy of a multi-way SVM.

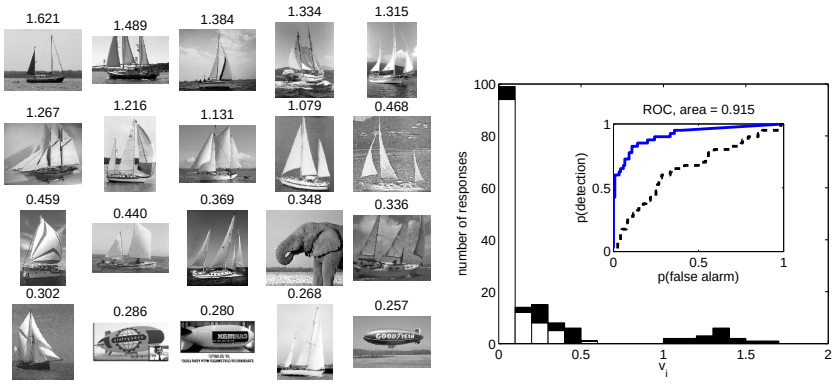
One surprising aspect of these results was the excellent performance in experiment (D), the 5-way face discrimination task which we initially tried as a presumably more difficult test of our methods. While the distinction between different faces is clearly more subtle than the distinction between categories, there is also less within-category variation in the face images than in the images from other categories, so different images of the same individual are likely to be tightly clustered in feature space. From this we see that the within-class homogeneity drives classification accuracy as much as the inter-class separation. Experiment (D) also highlights the importance of the statistics of the input set to the representation learned. In experiments (A) and (B), faces were present often in the inputs, but no particular individual was present often. In this case we obtain a representation for "face," but no individuation within that class. In experiment (D), particular individuals were present often, giving the network enough information to identify multiple individuals and represent them separately.

The seemingly poor results from experiment (C) still occur in the context of units that show very clean selectivity for each category. However, in each case the units responded strongly only to a *subset* of the category in question. Figure 2 gives an example of such a unit which responded selectively to some but not all of the ketch images.

5 Conclusions and Future Work

We here demonstrated a system that is able to group unlabeled images into appropriate categories through unsupervised learning on image features. This model has at its core the notion that underlying the high-dimensional vector of features from the model is a sparse set of causes, and that these causes can be uncovered by optimizing a sparse generative model of the inputs. This model performs quite well on benchmark image classification tasks despite being both entirely unsupervised and motivated primarily by the relevant biology rather than by optimizing machine vision performance. This model has the further important feature that it is not necessary to specify *a priori* the number of categories to search for, except of course to ensure that enough output units are available to represent all the input categories.

Many open questions remain. The simplest is how well this technique scales to larger



(a) (b)

Figure 2: Responses of a ketch unit from experiment (C). (a): images that evoked the top responses, with the activation level above each image. Every 2^{nd} image omitted for clarity. (b): response histogram. *x*-axis is the activation level; *y*-axis is the number of test images (160 total) evoking a response at that level. Responses to ketches in black; responses to all other images in white. Inset: ROC curve. Solid line is ROC curve for this unit, dashed line is ROC curve for best principal component. ROC equal error accuracy with respect to all ketches was 85%.

numbers of categories and categories that resemble one another more closely or are more diverse. It remains to be seen whether the feature set used in this investigation is sufficient to discover more (or more similar) categories in this unsupervised setting, or if the underlying visual system model itself is sophisticated enough to scale regardless of the number of features used. Our immediate future work will investigate this scalability.

Acknowledgments. We thank Thomas Serre and Minjoon Kouh of MIT for providing the visual system model used here as well as assistance with its operation, and Richard Murray, Jerry Marsden, and Pietro Perona at Caltech and Bruno Olshausen at Berkeley for valuable feedback. This work was funded by a Fannie and John Hertz Foundation Fellowship (to S.W.), as well as by grants from the ONR, NIMH, NSF, and DARPA.

References

[1] K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, and M. Jordan. "Matching words and pictures". *Journal of Machine Learning Research*, 3:1107–1135, 2003.

[2] A.J. Bell and T.J. Sejnowski. "An Information-Maximization Approach to Blind Separation and Blind Deconvolution". *Neural Computation*, 7:1129–1159, 1995.

[3] A.J. Bell and T.J. Sejnowski. "The 'Independent Components' of Natural Scenes are Edge Filters". *Vision Research*, 37(23):3327–3338, 1997.

[4] R. Fergus, P. Perona, and A. Zisserman. “Object Class Recognition by Unsupervised Scale-Invariant Learning”. In *Proc. CVPR*, 2003.

[5] G. Griffin, A.D. Holub, and P. Perona. “The Caltech 256”. Technical report, Caltech, 2006.

[6] G.E. Hinton and Z. Ghahramani. “Generative models for discovering sparse distributed representations”. *Phil. Trans. R. Soc. Lond. B*, 352:1177–1190, 1997.

[7] Y. Li, A. Cichocki, and S. Amari. “Analysis of Sparse Representation and Blind Source Separation”. *Neural Computation*, 16:1193–1234, 2004.

[8] J. Mutch and D.G. Lowe. “Multiclass Object Recognition with Sparse, Localized Features”. In *Proc. CVPR*, 2006.

[9] B.A. Olshausen and D.J. Field. “Emergence of simple-cell receptive field properties by learning a sparse code for natural images”. *Nature*, 381:607–609, 1996.

[10] B.A. Olshausen and D.J. Field. “Sparse Coding with an Overcomplete Basis Set: A Strategy Employed by V1?”. *Vision Research*, 37(23):3311–3325, 1997.

[11] R. Quian Quiroga, L. Reddy, G. Kreiman, C. Koch, and I. Fried. “Invariant visual representation by single neurons in the human brain”. *Nature*, 435:1102–1107, 2005.

[12] M. Ranzato, C. Poultney, S. Chopra, and Y. LeCun. “Efficient Learning of Sparse Representations with an Energy-Based Model”. In *Advances in Neural Information Processing (NIPS 2006)*, 2006.

[13] M. Riesenhuber and T. Poggio. “Hierarchical models of object recognition in cortex”. *Nature Neuroscience*, 2(11):1019–1025, 1999.

[14] E.T. Rolls and M.J. Tovee. “Sparseness of the Neuronal Representation of Stimuli in the Primate Temporal Visual Cortex”. *Journal of Neurophysiology*, 73(2):713–726, 1995.

[15] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. “Object recognition with cortex-like mechanisms”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):411–426, 2007.

[16] T. Serre, L. Wolf, and T. Poggio. “Object recognition with features inspired by visual cortex”. In *Proc. CVPR*, 2005.

[17] J. Sivic, B.C. Russell, A.A. Efros, A. Zisserman, and W.T. Freeman. “Discovering Object Categories in Image Collections”. Technical Report MIT-CSAIL-TR-2005-012, MIT CSAIL, 2005.

[18] S. Waydo, A. Kraskov, R. Quian Quiroga, I. Fried, and C. Koch. “Sparse Representation in the Human Medial Temporal Lobe”. *Journal of Neuroscience*, 26(40):10232–10234, 2006.

[19] M. Weber, M. Welling, and P. Perona. “Towards Automatic Discovery of Object Categories”. In *Proc. CVPR*, 2000.

Improvement of Retrieval Speed and Required Amount of Memory for Geometric Hashing by Combining Local Invariants

Masakazu Iwamura, Tomohiro Nakai, and Koichi Kise
Osaka Prefecture University, Japan
{masa,kise}@cs.osakafu-u.ac.jp, nakai@m.cs.osakafu-u.ac.jp

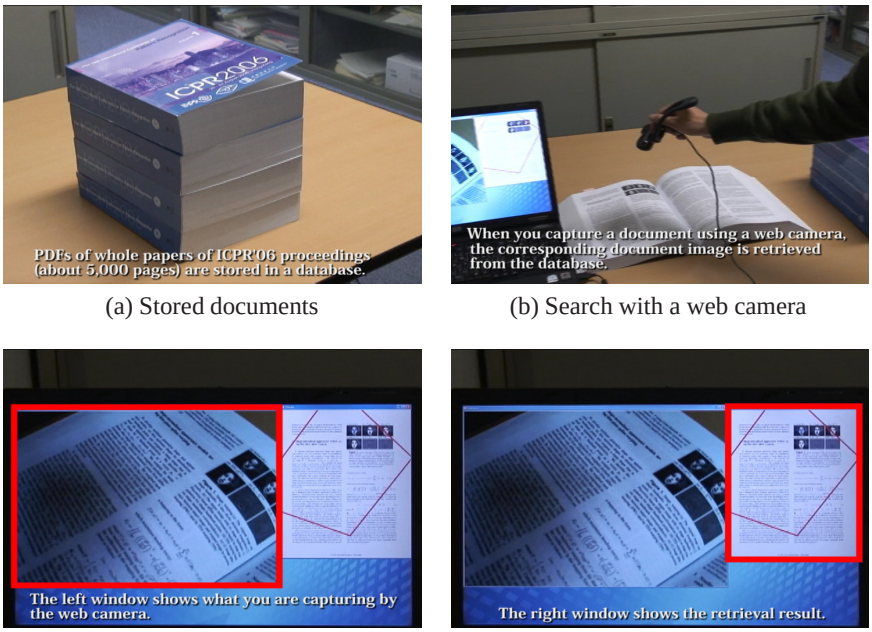
Abstract

The geometric hashing (GH) is a well-known model-based object recognition technique with good properties both in retrieval speed and required amount of memory. However, it has a significant weak point; as the number of objects increases, both retrieval speed and required amount of memory increase in the cubic, fourth or higher order. Recently, a new technique "locally likely arrangement hashing (LLAH)" whose computational cost is a linear order has been proposed. The objective of the current paper is to reveal how LLAH improves the performance. By comparing GH and LLAH, we describe four primary factors of the performance improvement.

1 Introduction

x $\boldsymbol{x}$ In the current paper, we discuss a problem to find the corresponding object from the database using feature points extracted from a query object. For the problem, rich descriptors such as SIFT descriptor [9] (a typical feature vector is 128 dimensions), PCA-SIFT descriptor [7] (typically 36 dimensions) and SURF descriptor [2] (typically 64 dimensions) are often used to describe the object. Since the descriptors are designed to be robust for noises such as change of intensity, rotation and scale, the corresponding object is retrieved by searching similar values of descriptors from the database. Popular searching methods include approximate nearest neighbor (ANN) [1], locality-sensitive hashing (LSH) [6, 3] and vocabulary tree [11]. However, distinctive rich descriptors are not always available. For example, a SIFT descriptor extracted from a texture type repeating pattern including a text region is not distinctive.

Thus, to the contrary to the rich descriptors, we discuss methods employing a poor descriptor whose feature is only *the location of the feature point*. This enables object recognition techniques to be widely applicable as mentioned below. However, this makes the problem difficult since retrieving the corresponding feature point in the database only with a single feature point is impossible. Focusing the arrangement of other feature points is necessarily required. Furthermore, if the query object image is taken from an oblique angle, it has undergone some geometric transformation. This makes the problem more difficult because the arrangement is no longer identical to the stored one in the database. Therefore, removing the effect of the geometric transformation is also required for precise retrieval.



(a) Stored documents (b) Search with a web camera

(c) Captured image in the left window (d) Retrieval result in the right window

Figure 1: Snapshots of a real-time image retrieval system [5] using LLAH [10]. The system is invariant against rotation, scaling, perspective distortion, occlusion and curvature of a page. The system works at around 7Hz for 5,000 pages, and the clock does not depend on the size of the database so much. (a) stored documents in the database, 5,000 pages in this demo. (b) a search scene with a 1.3M pixels web camera. (c) a screen shot including a captured image in the left window. (d) a screen shot including a thumbnail of the retrieved page and the corresponding region to the captured image drawn in a red rectangle in the right window.

The simplest way to resolve the problem is an exhaust search which requires the computational cost of $O(N!)$ for a search of an object including N feature points under perspective distortion. The cost is reduced by the geometric hashing (GH) down to $O(N^5)$ introducing an idea of geometric invariance [8, 16]. GH is known to be a general model-based object recognition method and widely used not only in computer vision, but also in many other domains including bioinformatics [12, 16]. However, GH still requires much processing time and large amount of memory, especially for a large number of objects. Many variants of GH have been proposed and some of them reduce the cost using probabilistic approach such as random reduction of feature points. However, probabilistic reduction of computational cost cannot avoid reduction of retrieval accuracy because computational cost and retrieval accuracy are in a trade-off relationship [4]. Therefore, it is quite difficult to apply GH and its variants to practical applications requiring quick response.

For the problem, we have proposed a new technique “locally likely arrangement hash-

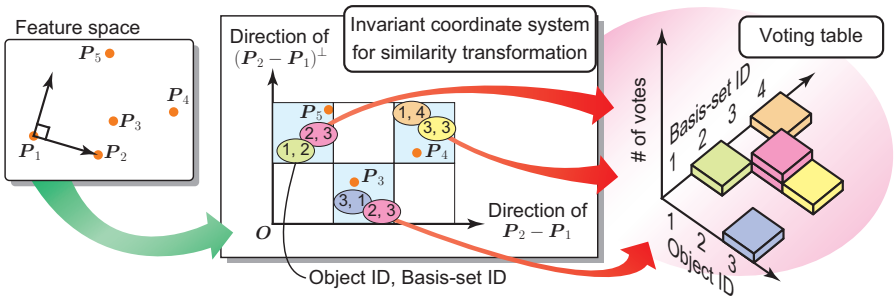


Figure 2: Retrieval process of the geometric hashing.

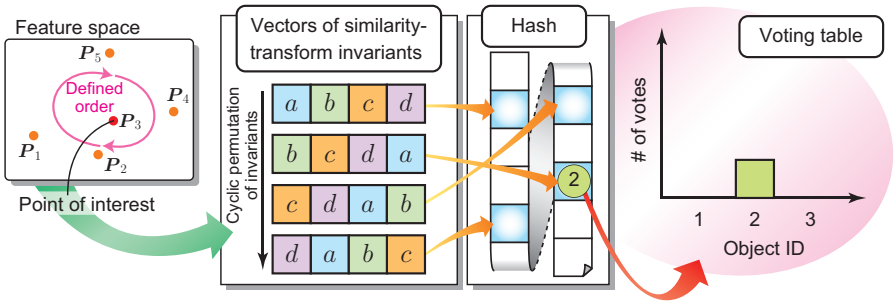


Figure 3: Retrieval process of LLAH. Invariants in the figure are $a = \frac{\|P_4 - P_1\|}{\|P_5 - P_1\|}$, $b = \frac{\|P_2 - P_5\|}{\|P_4 - P_5\|}$, $c = \frac{\|P_1 - P_4\|}{\|P_2 - P_4\|}$, and $d = \frac{\|P_5 - P_2\|}{\|P_1 - P_2\|}$, respectively.

ing (LLAH)" which outperforms GH in both processing time and required amount of memory [10]. LLAH requires the computational cost of $O(N)$ for a search. Due to its outstanding performance, we have applied it to a real-time image retrieval system [5] (see Fig. 1 for snapshots). In about 150 milliseconds, the system with 4GB memory can retrieve a corresponding image from 5,000 images, and find the corresponding region captured by a web camera. In spite of its outstanding performance, LLAH has not analyzed in detail. Therefore, the objectives of the current paper is to reveal where the outstanding performance of LLAH comes from. For the sake of the purpose, we compare GH and LLAH, and describe four primary factors of the performance improvement.

2 Geometric hashing and LLAH

2.1 Geometric hashing [8]

We explain the storage and retrieval processes of GH and LLAH in the case under a similarity transformation.

2.1.1 Storage process

GH describes the object which has undergone a certain geometric transformation using invariant coordinate systems. We explain the storage process of it. Though Fig. 2 is an illustration of the retrieval process, it will help understand the storage process because the most are common.

To begin with, feature vectors are extracted from the image of the object. Two of them are chosen, and a pair of bases is defined as shown in Fig. 2. In the figure, a basis $\boldsymbol{P}_2 - \boldsymbol{P}_1$ and its orthogonal one (denoted as $(\boldsymbol{P}_2 - \boldsymbol{P}_1)^{\perp}$) are defined. Then, the rest of the feature points ($\boldsymbol{P}_3$, $\boldsymbol{P}_4$ and $\boldsymbol{P}_5$) are projected to the invariant coordinate system spanned by the pair of bases ($\boldsymbol{P}_2 - \boldsymbol{P}_1$ and $(\boldsymbol{P}_2 - \boldsymbol{P}_1)^{\perp}$). The invariant coordinate system is divided (quantized) into subregions in advance. Thus, the object ID and a basis-set ID are stored into the each corresponding subregion.

The storage process finishes after the procedure above is carried out for all the invariant coordinate systems spanned by all the pairs of bases and for all the objects to be stored.

2.1.2 Retrieval process

We explain the retrieval process of GH with the illustration of Fig. 2. The initial phase of the retrieval process is almost the same as that of the storage one.

To begin with, feature vectors are extracts from the image of the object. Two of them are chosen, and a pair of bases is defined as shown in Fig. 2. Then, the rest of the feature points are projected to the invariant coordinate system spanned by the pair of bases. Each projected feature vector corresponds to a subregion of the invariant coordinate system. The votes for the corresponding pairs of the object ID and the basis-set ID are made.

The procedure above is carried out for all the invariant coordinate systems. The pair of the object ID and the basis-set ID with the highest vote determines the corresponding object. In the illustration of Fig. 2, the object #2 is retrieved because the combination of the object #2 and the basis-set #3 obtains the highest vote. The process can quit when the corresponding object is obviously determined.

2.2 LLAH [10]

2.2.1 Storage process

We explain the storage process of LLAH. Like Fig. 2, Fig. 3 is an illustration of the retrieval process. However, it will also help understand the storage process because the most are common.

After feature points are extracted, LLAH calculates feature vectors which describe the arrangements of the m nearest feature points of the feature point of interest. In Fig. 3, the point of interest is $\boldsymbol{P}_3$, and the $m\ (=4)$ nearest ones are $\boldsymbol{P}_1$, $\boldsymbol{P}_2$, $\boldsymbol{P}_4$, and $\boldsymbol{P}_5$. The feature vectors of $\boldsymbol{P}_3$ are calculated as follows. The m nearest points are ordered in clockwise rotation as $\boldsymbol{P}_5, \boldsymbol{P}_4, \boldsymbol{P}_2, \boldsymbol{P}_1, \boldsymbol{P}_5, \cdots$. With three points denoted as A, B and C, a similarity invariant $\frac{\mathrm{AC}}{\mathrm{AB}}$ is calculated, where AB stands for the line segment between A and B. Thus, a similarity invariant is calculated from a set of three points. By sliding the points to regard A, B and C in clockwise rotation, $\binom{m}{3}\ (=4)$ similarity invariants are calculated (i.e. a, b, c and d in Fig. 3). By combining $\binom{m}{3}$ invariants using clockwise

rotation, m vectors are created (i.e. $abcd$, $bcda$, $cdab$ and $dabc$ in Fig. 3). In the storage process, one of them is chosen arbitrarily. Then, a hash value is calculated from the chosen feature vector. Finally, the object ID is stored into the corresponding cell of the hash table in the hash value.

The storage process finishes after the procedure above is done selecting each feature point of all the objects to be stored as a point of interest.

For simplicity, in the above explanation, we omit an important factor of LLAH. Because LLAH creates the feature vectors by combining the information of coordinates of m neighbors (feature points), it is less robust than GH to the appearance and disappearance of feature points. In order to overcome the weakness, LLAH once chooses n ($\geq m$) neighbors, and then chooses m neighbors from the n neighbors. This makes it possible to perform a robust retrieval with a small increase of computational cost. This is described in Sec. 3.2.4.

2.2.2 Retrieval process

We explain the retrieval process of LLAH. The initial phase of the retrieval process is almost the same as that of the storage one.

After feature points are extracted, LLAH calculates feature vectors of the point of interest as in the storage process. With the illustration of Fig. 3, m vectors are created as in the storage process (i.e. $abcd$, $bcda$, $cdab$ and $dabc$). Though only one of them is used in the storage process, all of them are used in the retrieval process. Hash values are calculated from the feature vectors, and votes for the object IDs are made [1].

The procedure above is done selecting each feature point as a point of interest. The object ID with the highest vote determines the corresponding object. In the illustration of Fig. 3, one vote for the object #2 and three votes for *nothing* are made. Thus, the object #2 is retrieved. The vote for nothing is caused by an empty cell. The empty cells appear because a high-dimensional feature vector is employed to calculate a hash value described in Sec. 3.2.3.

Like GH, the process can quit when the corresponding object is obviously determined, for example, in the case that the difference between the highest vote and the second highest one is large.

3 Primary factors of the performance improvement of LLAH

Computational cost (processing time) and required amount of memory of LLAH decrease greatly compared to those of GH. In this section, we discuss four primary factors of it.

3.1 Problems of geometric hashing

Before discussing LLAH, we discuss the problems of GH first. In many practical situations, GH is unusable because it requires large amount of computation and memory. We

[1] Storing not only object IDs but also feature point IDs enables us to know the correspondence of feature points between the query and stored images with a slight increase of computational and memory resources. We have applied this to a real-time document image retrieval system [5].

consider three causes.

For the preparation, let N be the average number of feature points in an image. M be the number of stored images. Let b be the number of bases used in GH. b is determined by the class of invariants, i.e. $b = 2$ for a similarity transformation, $b = 3$ for an affine transformation, and $b = 4$ for a perspective transformation.

The first cause is that the computational cost based on the number of feature points (N). The original GH [8] requires the computational cost of $O(N^{b+1}M)$ in the storage process and $O(N^{b+1})$ in the retrieval process. Required amount of memory is the same as the required computational cost in the storage process. Therefore, even a modern computer cannot handle only several hundred points per image in practical time. Some variants of GH which reduce computational cost by thinning out feature points or basis-pairs to process probabilistically have been proposed. However, such probabilistic reduction of computational cost cannot avoid reduction of retrieval accuracy because computational cost and retrieval accuracy are in a trade-off relationship [4].

The second cause is hash collisions. The number of stored entries (sets of an object ID and a basis-set ID) in the hash table of GH is $N^{b+1}M$. If an invariant coordinate system is divided (quantized) into k bins per axis, k^b subregions per coordinate system exist. Therefore, the number of entries in a subregion is $N^{b+1}M/k^b$ in average. This is the same as the number of collisions. This means that one hash value causes as many as $N^{b+1}M/k^b$ votes! Since it is not easy to imagine how much $N^{b+1}M/k^b$ is, let us calculate it when $N = 100$, $M = 100$, $b = 3$ and $k = 10$. We can find the answer is 10^7! Even if k is changed to 100, the value is still as much as $10,000$. This is a major reason that GH cannot employ even moderate size of N and M in practice. A simple solution to avoid such enormous computation cost is to increase k. However, this also increases the possibility that a different bin is selected by a same feature vector by a noise because the hash table is divided finer. As the result, retrieval accuracy is also reduced.

The third cause is selection cost of the highest vote. GH has N^bM bins[2] in the voting table. In order to find the bin with the highest vote, all the bins have to be examined. Therefore, the computational cost of $O(N^bM)$ is required.

3.2 LLAH as the outcome of step-by-step improvements on the original geometric hashing

In order to reveal the relevance and difference between GH and LLAH, we regard LLAH as the outcome of step-by-step improvements on the original GH. That is, four improvements on the original GH derive LLAH and resolve the problems mentioned in Sec. 3.1.

3.2.1 Introduction of point of interest and m neighbors

The original GH uses all feature points in an image. We reduces the computational cost by reducing them. As mentioned in Sec 3.1, probabilistic reduction of feature points and basis-pairs cannot avoid retrieval accuracy. Thus, we define each feature point as a point of interest. Besides, only m neighbors of the point of interest are used for calculation. They reduce computational cost as much as $O(m^bNM)$ in the storage process and $O(m^bN)$ in the retrieval process.

[2] N^bM comes from the number of the object IDs (M) and the number of the basis-set IDs (N^b).

Note that the most important matter is that the same m points are obtained in the storage and retrieval processes. Obviously, if the image is taken from an oblique angle, retrieval accuracy can decrease due to change of m neighbors. This problem is discussed in Secs. 3.2.4.

3.2.2 Non-probabilistic reduction of invariants

We resolve the first problem mentioned in Sec. 3.1 by reducing the number of invariants. LLAH attaches great importance to select the same feature points to calculate invariants in reproducible manner. This enables to decreases computational cost without reducing retrieval accuracy. In order to realize it, we introduce clockwise order because

> clockwise order of feature points around the point of interest is invariant to geometric transformations.

By introducing the order, selectability of a sequence of feature points is greatly reduced. As an example, let's think about choosing two out of three points. Let A, B and C be the three points. Without introducing order, there are six choices such as AB, AC, BA, BC, CA and CB. However, by introducing an order like A $\rightarrow$ B $\rightarrow$ C which means AB, AC and BC are allowed to choice, but neither BA, CA nor CB is not. This decreases six choices to three. And, we can employ the same three choices with the order (reproducibility).

We explain the procedure of calculating invariants in *order-introduced* LLAH. Firstly, m neighbors of the point of interest are selected, and ordered in clockwise. Note that the beginning of the ordered points can be arbitrarily chosen because we handle only m points; Testing m possible beginning points increases the computational cost only m times and a constant growth of the cost is trivial compared to an exponential growth. Secondly, $b+1$ points are chosen form the m points keeping the order. Thirdly, an invariant is calculated with the arrangement of the $b+1$ points. Since a $(b+1)$-combination taken from the m points is $\binom{m}{b+1}$, $\binom{m}{b+1}$ invariants are calculated for a point of interest. The above procedure is repeated by selecting one of N points in an image as a point of interest. Thus, we have $\binom{m}{b+1}N$ invariants for an image. Finally, the computational cost in the storage process is $O\left(\binom{m}{b+1}NM\right)$. This is because $\binom{m}{b+1}N$ invariants are calculated for each of M objects. The cost in the retrieval process is $O\left(\binom{m}{b+1}mN\right)$. This is because $\binom{m}{b+1}N$ invariants are calculated for each of m beginning points.

For comparison, we explain how many invariants GH calculates. GH calculates invariants by projecting feature points to an invariant coordinate system since the coordinates on the invariant coordinate system are invariants. Since the dimensionality of the coordinate system is 2, two invariants for each point are calculated. Thus, the number of calculated invariants is given as

$$\text{(The number of basis-sets)} \times \text{(the number of projected points)} \times 2.$$

The number is the same order as the computational cost of a retrieval. That of the original GH is $O(N^b) \times (N-1) \times 2 = O(N^{b+1})$ and that of the method introduced in Sec. 3.2.1 is $O(Nm^{b-1}) \times (m-b+1) \times 2 = O(m^b N)$. This means that the computational cost of GH and LLAH is proportional to the number of calculated invariants. Therefore, we confirmed LLAH reduced the cost since LLAH reduced the number of calculated invariants.

3.2.3 Introduction of high-dimensional feature vector

We resolve the second problem mentioned in Sec. 3.1, which is the collision problem. The problem that many collisions occur in GH comes from low discrimination ability. As mentioned in Sec. 3.1, the number of stored entries is as many as $N^{b+1}M$, while the size of the hash table (the number of subregions) is only k^b. Thus, as a solution, LLAH enhances the discrimination ability by enlarging the size of the hash table. For the sake of that, the discrimination ability of invariants is also enhanced. Thus, LLAH combines $\binom{m}{b+1}$ invariants and creates a $\binom{m}{b+1}$-dimensional feature vector. The alignment is determined by use of clockwise order. Since each invariant is quantized into k discrete values, the size of the hash table is as much as $k^{\binom{m}{b+1}}$ at most.

We show an actual data that a large size hash table reduces collisions. In the case that LLAH is applied to a document image retrieval task[3], very sparse hash tables were obtained: only 2.95% of hash bins were nonempty for 1,000 pages (images), and 19.7% for 10,000 pages. In the nonempty bins, the average number of collisions was less than two.

Such a sparse hash table contributes to not only reduction of processing time, but also robust retrieval. As mentioned above, combining the information of feature points requires a complete match of $\binom{m}{b+1}$ discrete values of vector elements. This can cause failure of a vote because the probability two feature vectors match is much lower than the one that two elements of vectors match. Even so, if the hash table is sparse, most of wrong hash values do not harm due to empty bins.

As a side effect of combining invariants, we have resolved the third problem mentioned in Sec. 3.1. Due to high discrimination ability of a feature vector, the basis-set IDs are no longer required for discrimination. Thus, the form of the voting table has been changed into the one in Fig. 3. This reduces the computational cost of finding the bin with the highest vote from $O(N^bM)$ down to $O(M)$.

3.2.4 Robustness by using "m neighbors from n neighbors rule"

As mentioned before, a feature vector is weak to disturbances such as the appearance and disappearance of feature points, and change of skew angle of an image. In order to overcome the weakness, LLAH once chooses n ($\geq m$) neighbors, and then chooses m neighbors from the n neighbors. We call this "m neighbors from n neighbors rule". The rule creates $\binom{n}{m}$ possible choices of m neighbors. By employing all of them, $\binom{n}{m}$ times of feature vectors are created. This makes it possible to perform a robust retrieval because the probability that the same m feature points (i.e., a feature vector) are chosen in the storage process and the retrieval process is not 0 even if up to $n-m$ points are lost.

Introducing "m neighbors from n neighbors rule" changes the computational cost of LLAH to $O\left(\binom{n}{m}\binom{m}{b+1}NM\right)$ in the storage process, and $O\left(\binom{n}{m}\binom{m}{b+1}mN\right)$ in the retrieval process. Though the rule increases computational cost $\binom{n}{m}$ times, a good choice of n and m does not increase it so much. For example, in the case of $n=8$ and $m=7$, $\binom{8}{7}=8$, and in the case of $n=10$ and $m=8$, $\binom{10}{8}=45$.

[3] An affine invariant was used. The size of the hash table was approximately 2^{27}. $n=7$, $m=6$, $k=15$ were used for parameters.

4 Discussion

LLAH combines the information of feature points to calculate invariants. There are some methods which combines the information of feature points. In this section, we compare LLAH and them to clear the novelty of LLAH.

There are some improved methods of GH which employ other primitive features than a feature point: a line segment [15] and a chain of connected line segments [13]. The latter reduces the computational cost more; [13] achieved $O(N)$ for N line segments. However, a chain of connected line segments is not available for most target objects. The advantage of LLAH is ability to combine discontiguous feature points in a defined order.

For a robust match of local features, [14] introduces a geometric constraint such that angles between feature points should be in a range, though it does not use GH. Though the constraint improves discrimination ability, the angles change under a perspective and an affine transformation. To the contrary, the constraint of LLAH (i.e., order) is invariant even if the transformation is a perspective transformation.

5 Conclusion

In this paper, we described the important factors of improving performance of the geometric hashing (GH) in both retrieval speed and required amount of memory. For the sake of it, we compared GH and locally likely arrangement hashing (LLAH). Consequently, we obtained four primary factors: (1) introduction of point of interest and m neighbors, (2) non-probabilistic reduction of invariants, (3) introduction of high-dimensional feature vector, and (4) introduction of "m neighbors from n neighbors rule." The most important one is to use non-probabilistic selection of feature points because probabilistic one cannot escape from the trade-off relationship between computational cost and retrieval accuracy.

In another aspect, the contribution of this paper is the first detailed analysis of LLAH. We pointed out that LLAH breaks the thetrade-off relationship and resolves a collision problem of hashing.

Future work includes an evaluation of robustness of LLAH against disturbances such as the appearance and disappearance of feature points.

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] 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.

[2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. In *LNCS (Proc. ECCV'06)*, volume 3951, pages 404–417, May 2006.

[3] Mayur Datar, Piotr Indyk, Nicole Immorlica, and Vahab S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proc. 20th SCG*, pages 253–262, 2004.

[4] Michael Hoffman and Michael Lindenbaum. Some tradeoffs and a new algorithm for geometric hashing. In *Proc. ICPR'98*, pages 1700–1704, 1998.

[5] http://imlab.jp/LLAH/.

[6] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proc. 30th ACM STOC*, pages 604–613, 1998.

[7] Yan Ke and Rahul Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *Proc. CVPR'04*, volume 2, pages 506–513, 2004.

[8] Yehezkel Lamdan and Haim J. Wolfson. Geometric hashing: a general and efficient model-based recognition scheme. In *Proc. ICCV'88*, pages 238–249, 1988.

[9] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Proc. ICCV'04*, 60(2):91–110, 2004.

[10] Tomohiro Nakai, Koichi Kise, and Masakazu Iwamura. Use of affine invariants in locally likely arrangement hashing for camera-based document image retrieval. In *LNCS (Proc. DAS'06)*, volume 3872, pages 541–552, February 2006.

[11] David Nistér and Henrik Stewénius. Scalable recognition with a vocabulary tree. In *Proc. CVPR'06*, volume 2, pages 2161–2168, 2006.

[12] Ruth Nussinov and Haim J. Wolfson. Efficient detection of three-dimensional structural motifs in biological macromolecules by computer vision techniques. In *Proc. Nat'l Acad. Sci. U. S. A.*, volume 88, pages 10495–10499, 1991.

[13] Charles A. Rothwell, Andrew Zisserman, David A. Forsyth, and Joseph L. Mundy. Using projective invariants for constant time library indexing in model based vision. In *Proc. BMVC'91*, 1991.

[14] Cordelia Schmid and Roger Mohr. Local grayvalue invariants for image retrieval. *IEEE Trans. PAMI*, 19(5):530–535, May 1997.

[15] Franc C. D. Tsai. Robust affine invariant matching with application to line features. In *Proc. CVPR '93*, pages 393–399, June 1993.

[16] Haim J. Wolfson and Isidore Rigoutsos. Geometric hashing: an overview. *IEEE Comp. Sci. and Eng.*, 4(4):10–21, 1997.

Object Detection Using A Shape Codebook

Xiaodong Yu, Li Yi, Cornelia Fermuller, and David Doermann

Institute for Advanced Computer Studies
University of Maryland, College Park, MD 20742 USA
{xdyu,liyi,fer,doermann}@umiacs.umd.edu

Abstract

This paper presents a method for detecting categories of objects in real-world images. Given training images of an object category, our goal is to recognize and localize instances of those objects in a candidate image.

The main contribution of this work is a novel structure of the shape codebook for object detection. A shape codebook entry consists of two components: a shape codeword and a group of associated vectors that specify the object centroids. Like their counterpart in language, the shape codewords are simple and generic such that they can be easily extracted from most object categories. The associated vectors store the geometrical relationships between the shape codewords, which specify the characteristics of a particular object category. Thus they can be considered as the "grammar" of the shape codebook.

In this paper, we use Triple-Adjacent-Segments (*TAS*) extracted from image edges as the shape codewords. Object detection is performed in a probabilistic voting framework. Experimental results on public datasets show performance similiar to the state-of-the-art, yet our method has significantly lower complexity and requires considerably less supervision in the training (We only need bounding boxes for a few training samples, do not need figure/ground segmentation and do not need a validation dataset).

1 Introduction

Recently, detecting object classes in real-world images using shape features has been explored in several papers. Compared to local features such as SIFT [10], shape features are attractive for two reasons: first, many object categories are better described by their shape than texture, such as cows, horses or cups; second, for objects with wiry components, such as bikes, chairs or ladders, local features unavoidably contain large amount of background clutter [1, 13]. Thus shape features are often used as a replacement of, or complement to local features [2, 6, 17].

One practical challenge for shape features is that they are less discriminative than local features . To overcome this limitation, several methods have been proposed to use a shape codebook for object detection [4, 16, 21]. Inspired by these works, we propose a new structure of the shape codebook for object detection in this paper. In the shape codebook, the shape codewords should be simple and generic such that they can be reused in different object categories. The geometrical relationships between the shape codewords specify the

characteristics of a particular object category. Thus they can be viewed as the "grammar" of the shape codebook.

In this paper, we explore a local shape feature proposed by Ferrari *et al*, [4] as the shape codeword and use the *Implicit Shape Model* [7, 8] to define the shape grammar. The shape feature is formed by chains of k connected, roughly straight contour segments (kAS). In particular, we use $k = 3$, which is called Triple-Adjacent-Segments (*TAS*). A *TAS* codebook entry consists of two components. (1) A prototype *TAS* that represents a group of similiar *TAS*s, which is called *TAS* codeword. (2) a group of vectors specifying the associated object centroids, and encode the shape grammar. During detection, we match each *TAS* from the test image to the codebook. When an entry in the codebook is activated, it casts votes for all possible object centroids based on the associated vectors. Finally, candidate object centroids are detected as maxima in the continuous voting space using Mean-Shift Mode Estimation. The object boundary is then refined as the enclosure of the matched *TAS*s associated to the detected object centroid.

The main contributions of this work are:

1. We propose a two-layer structure of the shape codebook for object detection. Simple and generic shape features are used as shape codewords and geometrical constraints are used as the shape grammar. Since the shape codewords are not designed for specific object classes (e.g., cows, horses, cars), they only need to be learned once. Then they can be used in all object categories.

2. We seperate the procedures of learning shape codewords and building shape grammar. With a set of learned shape codewords, shape grammar can be learned for a new object category using a simple nearest neighbor rule. This method significantly reduces the complexity of the codebook and makes our algorithm more flexible.

The paper is structured as follows. The next section reviews related work. The proposed algorithm is described and evaluated in Section 3 and Section 4 respectively. Finally, Section 5 presents conclusions and future work.

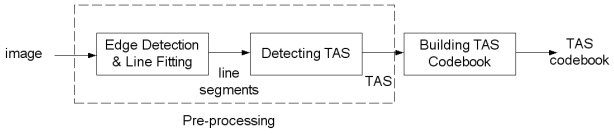
2 Related Work

Codebook of local features for object categorization and detection: The idea of learning a codebook for object categorization and detection has widely been used in approaches using local features in recent years [2, 3, 5, 11, 15, 19, 22, 24]. One of the key differences between these algorithms lies in the way the geometric configuration of parts in an object being exploited. The simple "bag-of-words" model is used in [5, 15, 24], where geometrical constraints among visual words are discarded. Loose spatial constraints are used in [22] to detect the co-occurence of pairs of visual words within a local spatial neighborhood. A slightly tighter spatial constraint called "spatial weighting" is decribed in [11], where the features that agree on the position and shape of the object are boosted and the background features are suppressed. Russell *et al* [19] encode the spatial relationship among visual words from the same object using segmentation information. Fergus *et al* [2] adopt a parameterized geometric model consisting of a joint Gaussian over the centroid position of all the parts. Translation and scale information is explicitly built in a pLSA model in [3], and clear improvement using this model is demonstrated on object classes with great pose variability.

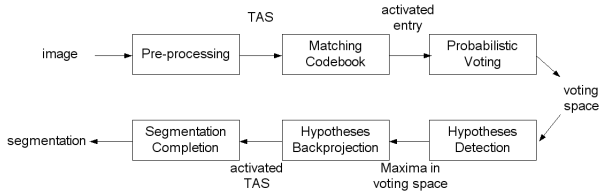
Codebook of shape features for object categorization and detection: The idea of learning a codebook has also been explored for shape features [4, 6, 9, 14, 16, 18, 21]. The different approaches employ diverse methods for building the shape codebook and using the geometrical constraints. Mori *et al* [14] quantize shape context vectors into a small number of canonical shape pieces, called *shapemes*. Liu *et al* [9] apply the "bag-of-words" model to 3D shape retrieval. Neither algorithm stores the spatial information. Kumar *et al* cluster outlines of object parts into a set of exemplar curves to handle variability in shape among members of an object class [6]. A pictorial structure is employed to represent the spatial relationship between parts of an object. Opelt *et al* [16, 18] build a codebook for class-discriminative boundary fragments and use boosting to select discriminative combinations of boundary fragments to form strong detectors. Similarly, a fragment dictionary is built by Shotton *et al* [21]. The differences between them are: the former requires no segmentation mask while the latter does; the former uses the spatial relationship between the boundary segments in a model similar to Leibe's approach [7], while the latter uses grids. Ferrari *et al* [4] build a codebook of *k*AS using the clique-partioning approximation algorithm. Compared to the codebooks used in [6, 16, 18, 21], the *k*AS codebook is generic and not designed for specific object classes (e.g., cows, horses, cars). Thus, once a codebook for a particular *k* has been learned, it can be used in all object classes.

3 The Algorithm

In this section, we present the details of the proposed algorithm (Figure 1). First, the pre-processing steps are described in Section 3.1. Then we discuss the approach for building the *TAS* codebook in Section 3.2. Finally, Section 3.3 explains how to detect objects in a test image.



(a) Training



(a) Object Category Detection

Figure 1: An overview flowchart of the proposed algorithm.

3.1 Detecting and Comparing *TAS*s

The *TAS* is used as the shape feature in our work. It is a special case of the *k*AS, which is formed by a chain of *k* connected, roughly straight contour segments. It has been shown that *k*AS is very robust to edge clutter. Since only a small number ($k \leq 5$) of connected segments are used, kAS can tolerate the errors in edge detection to some extent. Thus *k*AS is an attractive feature compromising between information content and repeatability. In the work of [4], object class detection is implemented using a sliding window mechanism and the best performance is achieved when $k = 2$.

We choose $k = 3$ in this work. As *k* grows, *k*AS can present more complex local shape structures and becomes more discriminative but less repeatable. Ferrari *et al* [4] point out that *k*AS of higher complexity are attractive when the localization constraints are weaker, and hence the discriminative power of individual features becomes more important. In this work, since we do not apply explicit spatial contraints, such as dividing the sliding window into a set of tiles, it is appropriate to use a *k*AS of higher degree.

The procedure to detect *TAS*s is summarized as follows: first, we detect image edges using the Berkeley Segmentation Engine (BSE) [12]. The BSE supresses spurious edges and has a better performance than the Canny edge detector. Second, small gaps on the contours are completed as follows: every edgel chain c_1 is linked to another edgel chain c_2, if c_1 would meet c_2 after extending *n* pixels. Contour segments are fit to straight lines. Finally, starting from each segment, every triplet of line segments is detected as a *TAS*.

Let θ_i, $l_i = \|s_i\|$ be the orientation and the length of s_i, where s_i for $i = 1,2,3$ denote the three segments in a *TAS* P. Two *TAS*s P^a and P^b are compared using the following measure $D(a,b)$

$$D(a,b) = w_\theta \sum_{i=1}^{3} D_\theta(\theta_i^a, \theta_i^b) + \sum_{i=1}^{3} |log(l_i^a / l_i^b)|, \tag{1}$$

where $D_\theta \in [0,1]$ is the difference between segment orientation normalized by π. Thus the first term measures the difference in orientation and the second term measures the difference in length. A weight $w_\theta = 2$ is used to emphasize the difference in orientation because the length of the segment is often inaccurate.

3.2 Building the *TAS* codebook

Building the *TAS* codebook consists of two stages: learning *TAS* codewords and learning *TAS* grammar. They are discussed in Section 3.2.1 and Section 3.2.2 respectively.

3.2.1 Learning *TAS* codewords

The *TAS* codewords are learned from *TAS*s in a training image set. First, we compute the distance of each pair of training *TAS*s. Then, we obtain a weighted graph $G = (V,E)$, where the nodes of the graph are the training *TAS*s, and an edge is formed between every pair of *TAS*s. The weight on each edge, $w(a,b)$, is a function of the distance between two *TAS*s P^a and P^b,

$$w(a,b) = exp\left(-\frac{D(a,b)^2}{\sigma_D^2}\right), \tag{2}$$

where σ_D is set to 20 percent of the maximum of all $D(a,b)$. Then clustering the training *TAS*s is formulated as a graph partition problem, which can efficiently be solved using the Normalized Cut algorithm [20].

After obtaining the clustering results from the Normalized Cut algorithm, we select a *TAS* codeword, J_i, from each cluster i. The *TAS* codeword is selected as the *TAS* closest to the cluster center (i.e., it has the smallest sum of distances to all the other *TAS*s in this cluster). Each codeword J_i is associated with a cluster radius r_i, which is the maximum distance from the cluster center to all the other *TAS*s within this cluster.

Figure 2.a shows the 40 most frequent *TAS* codewords in the codebooks learned from 10 images in the Caltech motorbike dataset. We can observe that the most frequent *TAS* codewords have generic configurations of three line segments. Quantitatively, we compared the codewords from variant datasets and found 90% to 95% of the *TAS* codewords are similiar. This confirms that the *TAS* codebooks are generic. In the following experiments, we apply the codewords learned from the Caltech motorbike dataset to all datasets.

(a) (b) (c)

Figure 2: Examples of *TAS* codewords. (a) shows the 40 most frequent *TAS* codewords learned from 10 images in the Caltech motorbike dataset. (b) and (c) illustrate the 5 most frequent *TAS* codewords (the first column) and their associated members in the clusters for the Caltech motorbikes dataset and the cows dataset respectively.

3.2.2 Learning *TAS* Grammar

To learn the *TAS* grammar, we need training images with the object delineated by a bounding boxes. First, we apply the nearest neighbor rule to quantize the *TAS*s within the bounding boxes using the *TAS* codewords. Let's denote e_k a *TAS* and J_i the nearest neighbor in the codebook. The *TAS* e_k is quantized as J_i if $D(J_i,e_k) < r_i$. Figure 2.b and 2.c show the 5 most frequent *TAS* codewords in two datasets and their associated members in the cluster. We found that only less than 2% of the *TAS*s in all datasets can not be found in the *TAS* codewords learned from the motorbike dataset. This further confirms the generality of the *TAS* codebook.

The *TAS* grammar is defined using the *Implicit Shape Model* [7]. For the member *TAS*s in cluster i of size M_i, we store their positions relative to the object center $(v_m, m = 1,...,M_i)$. Thus, a codebook entry records the following information: $\{J_i; (v_m, m = 1,...,M_i)\}$. For simplicity, we might also use J_i to denote the codebook entry.

3.3 Detecting Object Category by Probabilistic Voting

The procedure for detecting object category is illustrated in Figure 1.b. First, we match each test image *TAS* e_k located at l_k to the codebook. A codebook entry J_i is declared

to be matched (activated) if $D(J_i, e_k) < r_i$. For each matched codebook entry J_i, we cast votes for possible locations of the object centers $(y_m, m = 1,...,M_i)$, where y_m can be obtained from l_k and v_m. Then, object category detection is accomplished by searching for local maxima in the probabilistic voting space after applying Parzen window probability density estimation. Formally, let x_n be a candidate position in the test image and $p(x_n)$ be the probability that object appears at position x_n. Candidate object centers x^* defined as follows,

$$x^* = arg\max_x \sum_{x_n \in W(x)} p(x_n), \tag{3}$$

where $W(x)$ is a circular window centered at x. The probability $p(x_n)$ is obtained by observing evidence e_k in the test image. Thus, conditioned on e_k, we marginalize $p(x_n)$ as follows

$$p(x_n) = \sum_k p(x_n|e_k)p(e_k). \tag{4}$$

Without any prior knowledge on $p(e_k)$, we assume it is uniformly distributed, i.e., $p(e_k) = 1/K$, where K is the number of *TAS*s in the test image.

Let $\mathbf{S}$ be the set of matched codewords, $p(x_n|e_k)$ can be marginalized on $J_i \in \mathbf{S}$

$$p(x_n|e_k) = \sum_{J_i \in \mathbf{S}} p(x_n|J_i, e_k)p(J_i|e_k) \tag{5}$$

$$= \sum_{J_i \in \mathbf{S}} p(x_n|J_i)p(J_i|e_k). \tag{6}$$

After matching e_k to J_i, the voting will be performed by members within J_i. Thus $p(x_n|J_i, e_k)$ is independent of e_k and Equation 5 can be reduced to Equation 6. In Equation 6, the first term is the probabilistic vote for an object position given an activated codebook entry J_i, and the second term measures the matching quality between J_i and e_k. The matching quality can be measured in a manner similar to Equation 2

$$p(J_i|e_k) \propto \exp\left(\frac{-D(e_k, J_i)^2}{r_i^2}\right). \tag{7}$$

For an activated codebook entry, we cast votes for all possible locations of the object centers y_m. Thus $p(x_n|J_i)$ can be marginalized as

$$p(x_n|J_i) = \sum_m p(x_n|y_m, J_i)p(y_m|J_i) \tag{8}$$

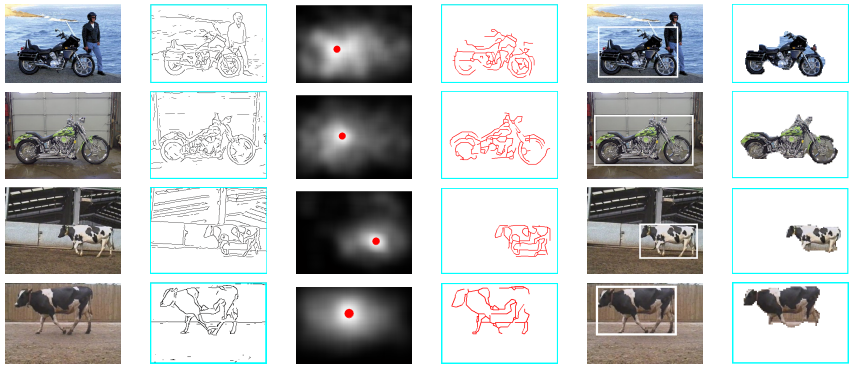
$$= \sum_m p(x_n|y_m)p(y_m|J_i). \tag{9}$$

Since the voting is casted from each individual member in J_i, the first term in Equation 8 can be treated as independent of J_i. Then Equation 8 is reduced to Equation 9. Without prior knowledge of y_m, we treat them equally and assume $p(y_m|J_i)$ is a uniform distribution, i.e., $p(y_m|J_i) = 1/M_i$.

The term $p(x_n|y_m)$ measures the vote obtained at location x_n given an object center y_m. Since we only vote at the location of possible object centers, we have $p(x_n|y_m) = \delta(x_n - y_m)$, where $\delta(t)$ is the Dirac delta function.

Combining the above equations, we can compute $p(x_n)$ from the evidence e_k located at l_k. In order to detect instances of the object category, we search for the local maxima

x^* in the voting space after applying Parzen window probability density estimation. The score of these candidates is defined as $\sum_{x_n \in W(x^*)} p(x_n)$. If this score is greater than a threshold t_{score}, we classify this image belonging to the training object category. To obtain a segmentation of the object instance, we find the test *TAS*s voting within $W(x^*)$ for an x^*. Then we obtain a smooth contour from these *TAS*s using the Gradient Vector Flow snake algorithm [23]. Also a bounding box is obtained in this procedure for each object instance. Figure 3 shows some detection examples for the Caltech motorbikes dataset and the cows dataset.



(a) (b) (c) (d) (e) (f)

Figure 3: Example detection results for the Caltech motorbikes dataset and the cows dataset. (a) The originial images. (b) The edge maps. (c) The voting spaces and detected centroids. (d) The backprojected *TAS*s. (e) The bounding box of the detected objects. (f) The segmentation

4 Experimental Results

In this section, we evaluate the performance of the proposed algorithm and compare it to the state-of-the-art algorithms that detect object categories using shape features. If a test image has a detection score greater than the threshold t_{score} and the overlap between the detected bounding boxes and the ground truth is greater than 50%, we consider the detection (localization) correct. By varying t_{score} we can obtain different recall/precision values. The performance is evaluated in terms of the Recall-Precision Equal Error Rate (RPC EER). All parameters are kept constant for different experiments.

The training data includes training images with bounding boxes annotating instances of the object class. Compared to the state-of-the-art, we require the least supervision. [16, 18] uses training image with bounding boxes and validation image sets that include both positive and negative images. [4] also requires negative images to train the SVM classifier. [21] requires segmentation masks for 10 positive training images plus a large amount of positive and negative images to train a discriminative classifier.

Cows Dataset: We use the same cow dataset as in [16] and compare to their results: 20 training images and 80 test images, with half belonging to the category cows and half

Table 1: Performance (RPC EER) depending on the number of training images with bounding boxes (N_{BB}) on the cows dataset and comparison to other published results.

	N_{BB}=5	N_{BB}=10	N_{BB}=20
Ours	0.93	0.95	0.96
Opelt [16]	0.91	0.95	1.00

Table 2: Performance (RPC EER) on the cups dataset and comparison to other publised results. N_{BB} is the number of training images with bounding boxes; N_V is the numbers of validation images.

	N_{BB}	N_V	RPC EER
Ours	16	-	0.841
Opelt [16]	16	30	0.812

to negative images (Caltech airplanes/faces). But we do not use the validation dataset while [16] uses a validation set with 25 positive/25 negative.

The performance is shown in Table 1. We also shows the variation in performance with the number of training images. The results show that our approach outperforms or performs as well as Opelt's when the number of training images is small ($N_{BB} = 5, 10$) but is outperformed when the number of training image is large ($N_{BB} = 20$). It shows that our approach is favorable when there are small number of training images available. The reason is that the *TAS* feature is very simple and generic. Thus only a few training images is sufficient to discover the statistical patterns in the training images. In comparison, Opelt's features are more complex and have more discriminative power for a particular object. Hence more training images are needed to fully exploit their advantages.

Cup Dataset: In this test, we evaluate our approach on the cup dataset used in [18]. We use 16 training images and test on 16 cup images and 16 background images. We do not use the validation set with 15 positive/15 negative, which is used in [18].

The performance is summarized in Table 2. It shows that we can achieve slightly better performance than Opelt's algorithm even we use less supervision in the training.

Caltech Motorbikes Dataset: In this test, we evaluate our algorithm using the Caltech motorbikes dataset [2]. Training is conducted on the first 10 images in this dataset. Testing is conducted on 400 novel motorbike images and 400 negative images from Caltech airplane/face/car rear/background images.

The experimental results are compared to other publised results on object localization in Table 3. We also compared the degree of supervision in the training in terms of the number of variant types of training images. It is shown that we can achieve performance compariable to Shotton's method but are slightly worse than Opelt's. This should be attributed to the class-discriminative contour segments used by Opelt *et al*.

Discussion: The advantage of the proposed method lies in its low complexity. The *TAS* codewords only need to be learned once. Thus the learning procedure for a new object category can be reduced to a simple nearest neighbor search for the training *TAS*s and the time-consuming clustering can be skipped. Furthermore, There are a limited number of possible configurations of three line segments. In our experiments, the *TAS* codebook has 190 entries. Ferrari *et al* [4] reported a *TAS* codebook with 255 entries

Table 3: Comparison of the proposed algorithm to other publised results on the Caltech motorbikes dataset. Column 2 through 5 are the numbers of variant types of training images: N_S for images with segmentations; N_U for images without segmentations; N_{BB} for images with bounding boxes; N_V for validation images.

	N_S	N_U	N_{BB}	N_V	RPC EER
Ours	-	-	10	-	0.921
Shotton [21]	10	40	-	50	0.924
Opelt [16]	-	-	50	100	0.956

because they used more complex descriptors. Nevertheless, the number of the shape codewords is bounded, rather than increasing linearly with the number of class categories as in the codebook used in [18, 16].

5 Conclusion

We have presented a two-layer structure of the shape codebook for detecting instances of object categories. We proposed to use simple and generic shape codewords in the codebook, and to learn shape grammar for individual object category in a seperate procedure. This method is more flexible than the approaches using class-specified shape codewords. It achieves similiar performance with considerable lower complexity and less supervision in the training. And thus it is favorable when there is a small number of training images available or the training time is crucial.

Currently we are investigating methods to combine several shape codewords in the voting. We will also try other clustering methods, e.g., k-means, aggolomerative clustering, etc., and compare the *TAS* codebooks to those used in this paper. Finally we plan further evaluation of the proposed method in more challenging datasets and over more categories.

References

[1] Owen Carmichael and Martial Hebert. Shape-based recognition of wiry objects. In *IEEE Conference On Computer Vision And Pattern Recognition*. IEEE Press, June 2003.

[2] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, volume 02, page 264, Los Alamitos, CA, USA, 2003. IEEE Computer Society.

[3] Robert Fergus, Fei-Fei Li, Pietro Perona, and Andrew Zisserman. Learning object categories from google's image search. In *ICCV*, pages 1816–1823, 2005.

[4] Vittorio Ferrari, Loic Fevrier, Frederic Jurie, and Cordelia Schmid. Groups of adjacent contour segments for object detection. *Technical Report*, 2006.

[5] Kristen Grauman and Trevor Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV*, pages 1458–1465, 2005.

[6] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Extending pictorial structures for object recognition. In *Proceedings of the British Machine Vision Conference*, 2004.

[7] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV'04 Workshop on Statistical Learning in Computer Vision*, pages 17–32, Prague, Czech Republic, May 2004.

[8] B. Leibe and B. Schiele. Interleaved object categorization and segmentation. In *British Machine Vision Conference (BMVC'03)*, pages 759–768, Norwich, UK, Sept. 2003.

[9] Yi Liu, Hongbin Zha, and Hong Qin. Shape topics: A compact representation and new algorithms for 3d partial shape retrieval. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2025–2032, Washington, DC, USA, 2006. IEEE Computer Society.

[10] D. Lowe. Distinctive image features from scale-invariant keypoints. 20:91–110, 2003.

[11] Marcin Marszalek and Cordelia Schmid. Spatial weighting for bag-of-features. In *CVPR*, pages 2118–2125, Washington, DC, USA, 2006. IEEE Computer Society.

[12] David R. Martin, Charless C. Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):530–549, 2004.

[13] K. Mikolajczyk, A. Zisserman, and C. Schmid. Shape recognition with edge-based features. In *Proceedings of the British Machine Vision Conference*, volume 2, pages 779–788, 2003.

[14] Greg Mori, Serge Belongie, and Jitendra Malik. Efficient shape matching using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1832–1837, 2005.

[15] David Nistér and Henrik Stewénius. Scalable recognition with a vocabulary tree. In *CVPR*, pages 2161–2168, 2006.

[16] Andreas Opelt, Axel Pinz, and Andrew Zisserman. A boundary-fragment-model for object detection. In *ECCV*, pages 575–588, 2006.

[17] Andreas Opelt, Axel Pinz, and Andrew Zisserman. Fusing shape and appearance information for object category detection. In *BMVC*, pages 117–127, Edinburgh, UK, 2006.

[18] Andreas Opelt, Axel Pinz, and Andrew Zisserman. Incremental learning of object detectors using a visual shape alphabet. In *CVPR*, pages 3–10, Washington, DC, USA, 2006. IEEE Computer Society.

[19] Bryan C. Russell, William T. Freeman, Alexei A. Efros, Josef Sivic, and Andrew Zisserman. Using multiple segmentations to discover objects and their extent in image collections. volume 02, pages 1605–1614, Los Alamitos, CA, USA, 2006. IEEE Computer Society.

[20] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[21] Jamie Shotton, Andrew Blake, and Roberto Cipolla. Contour-based learning for object detection. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, pages 503–510, Washington, DC, USA, 2005. IEEE Computer Society.

[22] Josef Sivic, Bryan Russell, Alexei A. Efros, Andrew Zisserman, and Bill Freeman. Discovering objects and their location in images. In *International Conference on Computer Vision (ICCV 2005)*, October 2005.

[23] C. Xu and J. Prince. Gradient vector flow: A new external force for snakes. In *Proceedings of Computer Vision and Pattern Recognition (CVPR '97)*, pages 66–71, San Juan, Puerto Rico, June 1997.

[24] Jianguo Zhang, Marcin Marszalek, Svetlana Lazebnik, and Cordelia Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *Int. J. Computer Vision*, 73(2):213–238, 2007.

Generic Object Recognition via Shock Patch Fragments

Özge C. Özcanlı and Benjamin B. Kimia
Division of Engineering
Brown University, Providence, RI, USA
{ozge, kimia}@lems.brown.edu

Abstract

We propose a new methodology to partition a natural image into regions based on the shock graph of its contour fragments. We show that these regions, or *shock patch fragments*, are often object fragments, thus effecting a partial segmentation of the image. We utilize shock patch fragments to recognize objects with dominant shape cues eliminating the need to segment out the entire object from the image first. Our preliminary results with minimal training are promising with respect to the state of the art recognition systems.

1 Introduction

There has been a paradigm shift in computer vision in the past decade moving away from relying on fully segmented images for recognition and other visual tasks, to using a collection of features that capture appearance and shape in a small area. The key point underlying this paradigm shift is the availability of a new generation of feature detectors such as Harris-Affine, Harris-Laplace and others [15] and a new generation of feature descriptors such as SIFT [13] and others [14] that are more stable to viewpoint variation, lighting change, *etc*. The main idea is that while these features may not remain present under all variations, given the sheer number of features, the common presence of a few discriminative features can discriminate between the presence or absence of a particular, previously observed object in an image.

In contrast to object instance recognition, generic object recognition, where the intra-category type variations must in addition be accounted for, has proven to be more challenging. Approaches to generic object recognition which are based on key feature detectors and descriptors span a continuum between two extremes. On one extreme, the spatial relationship between parts is represented such as in the constellation model [6, 4] and the k-fan model [3], and these are referred to as "part-based" models. On the other extreme which completely discards the spatial relationship, the approach relies on an unorganized collection of features which are coded in a lower dimensional vocabulary of visual words, or a codebook of appearance parts common to a collection of images, and are known as "bag of words" models. The former "part based" approach is faced with a combinatorial search arising from an exponential number of correspondences. The "bag of words" approach avoids the combinatorial difficulties [29, 5, 28, 35], but is more brittle to situations when some of the features have been removed, *e.g.*, due to partial occlusion. In approaches between these extremes, the geometric relationships between neighboring

features is modeled using specified geometric transformations [21, 22, 33], or using lose pairwise relationships [1, 34].

The success of the above approaches indicates the significance of the role of *appearance* in discriminating the presence or absence of objects in typical scenes where bottom-up segmentation could not conceivably segment figure from ground. However, these approaches are also limited in several significant ways. The chief drawback is that stable key features are not always available to the degree of abundance that the approach relies on. For example, the intensity variation in the interior of an object viewed in low lighting condition, *e.g.*, a cow at dusk/dawn, or equivalently against bright backgrounds, *e.g.*, a bird against a bright sky, is too limited to produce reliable key features. In this case the silhouette is a more reliable cue for recognition. As another example, objects especially man-made objects, may feature large homogeneous and therefore featureless areas. Similarly, cartoons, sketches, and line drawings which are readily recognizable would have no appearance-based key features. In these cases, the edge content is the sole information source for recognition. Finally, objects in low resolution imagery, *e.g.*, aerial video images of vehicles, where the total extent of the object is of the same order as that required for feature descriptors (25x25) [19], cannot be recognized using this approach.

A second short coming of the use of appearance-based key features for recognition is that the role of appearance may become severely diminished as the size of the database grows. This happens when the type variation increases the range of appearances on object category captures. For example, in recognizing bottles and cups, the surface markings are simply too varied to be useful [17]. In this case, the edge content of the silhouette and of the internal markings consistent across the category become the primary source of information for recognition.

A number of recent approaches to object recognition rely on the edge content of category, as represented in an unorganized edge map, or in a collection of curve fragments. As an example of use of edge maps in recognition, Belongie *et al.*'s [2] shape context approach assigns a signature to each edge representing the radial-polar histogram of other edges. This signature is sufficiently discriminative to enable correspondence and a similarity score after an image transformation. As an example of an approach relying on contour fragments, Nelson and Selinger [16, 24], motivated by the cubist approach to evoking the visual percept of form from a few fragmentary cues, modeled contour fragment maps by a collection of local context patches (21x21) which are normalized for size and orientation with respect to a centrally placed key curve. Fergus *et al.* [7] use segments of extended edge chains lying between bitangent points in their constellation model. Kumar *et al.* [12] used contours as a component in a graph-based pictorial structures. In the Boundary Fragment Model (BFM), a boundary fragment codebook is constructed by clustering those which are highly class-distinctive and predictive of the object centroid over a set of training data [17, 25].

A significant disadvantage of the above approaches is that either the relative spatial distribution of various contour fragments in an object is ignored altogether, or it is captured through the mediation of an object model, *e.g.*, requiring an object centroid. The lack of the relative spatial relationship among contour fragments restricts the discriminability of each fragment. The requirement of an object model to mediate the spatial relationship between fragments renders the approach sensitive to partial occlusion. For example, if only the head of a horse or a cow is visible, individual contour fragments for the head can match a large number of fragments from other objects, an effect which

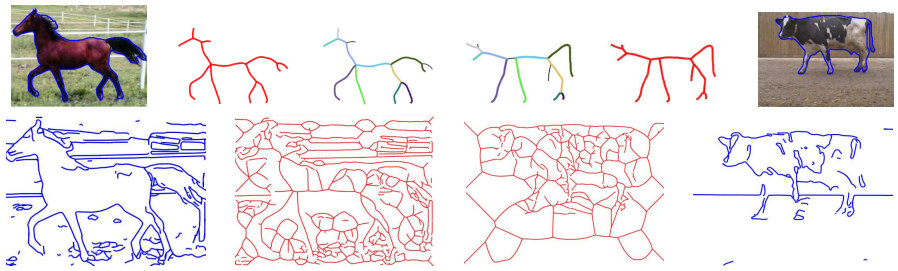


Figure 1: Top row illustrates how the shock graph of a horse is optimally transformed to the shock graph of a cow (colored edges are matching and the thinner black edges are editted out). While for segmented images this can be done with a polynomial time algorithm, the algorithm for matching the shock graphs in the bottom row is NP-complete (blue: boundaries, red: shock graphs).

becomes more significant as the size of the database increases. The spatial relationship among pairs of contour fragments on the head, on the other hand, make the joint-pair of fragments highly selective.

The main goal of this paper is to use the joint representation of a pair of contour fragments for recognition. The medial axis is a structure for the joint representation of pairs of contour fragments and our paper is focused on the use of this structure in the form of a shock graph as described below. The only previous work which takes advantage of pairs of contour fragments in such a "localized" sense is that of Jurie and Schmid [9] where edges are detected at multiple scales and annular regions are rated for the extent of significant non-accidental edge support on a wide range of angles around the region, see also [10]. The annular regions are localized over position and scale and used as distinctive and discriminative shape features. However, these shape features do not make use of the geometry of the curve fragments beyond the presence or absence in the small portions falling in the thin annular regions.

Our work builds on the success of shock graphs as a representation for generic object recognition from segmented images [23, 27]. Shock graph is a variant of the medial axis of the contour map of an image and it is obtained by viewing the medial axis as the locus of singularities (shocks) formed in the course of wave propagation (grass-fire) from boundaries [11, 26, 32]. The resulting shock graph is a richer descriptor of the contour map than the medial axis graph and it is a good intermediate representation since its nodes and edges signify presence of contour pairs and triplets, gaps and T-junctions. Loops in the graph signify groups of edges. See Figure 2c and 3c for example shock graphs of two contour sets. The use of the shock graph captures much of the intra-class object variability since articulation and metric variations in the part shape often leave the structure intact, while partial occlusion only affects parts of it. Those changes that lead to structural changes in the shock graph are captured in the context of considering deformation paths encoded by shock transitions. The precision-recall rates for large number of categories is excellent [23]: for a database of 1032 shapes roughly organized in 40 categories, the leave-one-out recognition rate is at 97% and drops to 82% for the last member of the category.

Generalizing the above approach from recognition of objects in segmented images to those in real images requires first that edge maps be represented by a shock graph, and furthermore that perceptual grouping operations, like removing an edge from the map or

closing a gap, be represented as a sequence of shock transitions, *e.g.* as described in [30]. The key difficulty is that the shock graph which for segmented images is a tree and which therefore leads itself to a polynomial-time edit distance algorithm, Figure 1, is no longer a tree due to the presence of spurious edges and gaps. Thus, matching two shock graphs faces combinatorial explosion in the search space and becomes NP-complete. This is very much similar to the combinatorial search space of constellation models, which limits its use to simpler models. In a similar vein, the complexity of the shock graph edit-distance algorithm for edge maps in real images motivates the recognition of smaller portions of objects, or object fragments. Our approach, therefore, probes the presence of an object in a limited portion of the shock graph, as determined by a collection of subgraphs. Each shock subgraph models a patch of the image which we refer to as a *shock patch fragment*. The recognition of these object fragments is the basis of our approach to object category recognition. While the use of shock object fragments based on "shock patch fragments" enables the use of both shape and appearance cues, we focus on shape features in this paper. However, it is not difficult to construe how a region descriptor of the sort used for key feature description can be used for shock patch regions to augment the shape aspects with appearance.

The paper is organized as follows. Section 2 describes how an edge map is processed to produce a collection of contour fragments from which a shock graph is obtained, Figures 2 and 3. In Section 3, we explain extraction of shock patches and show examples. In Section 4 we describe the procedure to match shock patch sets for object recognition and we report our results on object detection task in Section 5.

2 Contour Fragments

The success of any method based on shock graphs of curve fragments heavily depends on the reliability and stability of image contours. We now explain the processing stages of our approach: edge detection, edge linking, and perceptual grouping.

Edge Detection and Edge Linking: We use a pair of recently proposed edge detection and edge linking algorithms [31] which robustly extract well-localized sub-pixel edges and stably links these into curve fragments. For edge detection, it advocates the use of a third-order edge detector with an extremely low threshold, to get as many edges as possible so that the linking stage has enough options to choose from. Since the low threshold creates many spurious curve fragments, we prune these after linking by thresholding a measure consisting of both length and color contrast in the LAB space as used in [20],

$$C_a = 1 - e^{\frac{-||\mu_{R^+} - \mu_{R^-}||}{\gamma_{app}}} \tag{1}$$

where μ_{R^+} and μ_{R^-} are the mean colors of regions on either side of the curve fragment, and $||.||$ is the L_2 distance in R^3. We set $\gamma_a = 14$. If a color image is not available we find the L_1 distance of the appearance means. Figure 2b and 3b shows the curve fragments resulting from this process using a length threshold of 2 pixels and a color contrast threshold of 0.5 with a support region width of 5 pixels.

Gap Closure: The shock graph of the resulting curve fragments is computed using the method in [30], see Figures 2c and 3c. There are numerous gaps and spurious curve fragments which interfere with the process of forming shock segments correspond to object fragments. The shock graph provides a clue to the existence of these elements and

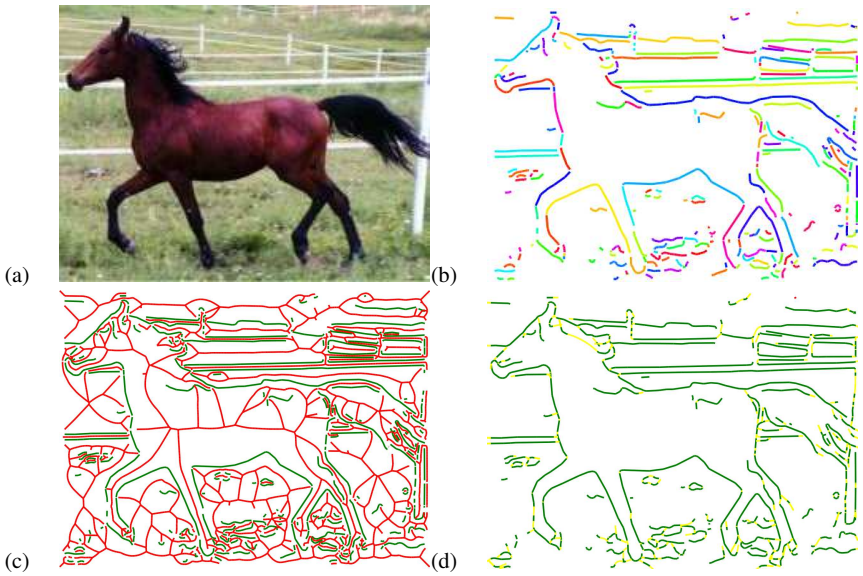


Figure 2: (a) An example image; (b) Curve fragments after pruning based on length and color contrast; (c) Shock graph of the curve fragment set (d) Curve fragments after the gap transform where the fragment set is shown in green and the gap completions in yellow.

transformations of it can be used to effect gap closure (*gap transform*) and the removal of spurious elements (*loop transform*). Specifically, observe that waves propagating radially from curve-ends meet, they form *degenerate* shocks in the shock graph, when they meet with normal waves propagating parallel to the contours, they form *semi-degenerate* shocks [8]. These edges signify gaps and possible T-junctions, respectively, in the curve fragment map of the image. See Figure 4a for an example of each kind. The gap transform is based on closing gaps and forming T-junctions by considering each case as rank-ordered by a measure reflecting both *(i)* good contour continuity and *(ii)* appearance discontinuity. The results are shown in Figures 4c, 2d and 3d.

3 Shock Patch Extraction

We now explore the notion of forming recognizable and stable image fragments which in effect are hypotheses for partial segmentations of the image. Assuming that the fragments have detectable boundaries, they must be anchored on curve fragments. Since a single curve fragment is not sufficiently distinctive, multiple contour fragments should be used to define image fragments. Since each pair of adjacent contour fragments give rise to a shock segment, selecting shock subgraphs provides a mechanism for selecting a group of curve fragments. Specifically, given a particular node in the shock graph, we traverse neighboring nodes in a depth-first manner to extract subgraphs at various depths. Since each shock segment typically describes a pair of curve fragments and the portion of the image in-between, we refer to this as a *visual fragment* Figure 5a, the shock subgraph describes an image fragment, which we refer to as the *shock patch fragment*, Figure 5e. The

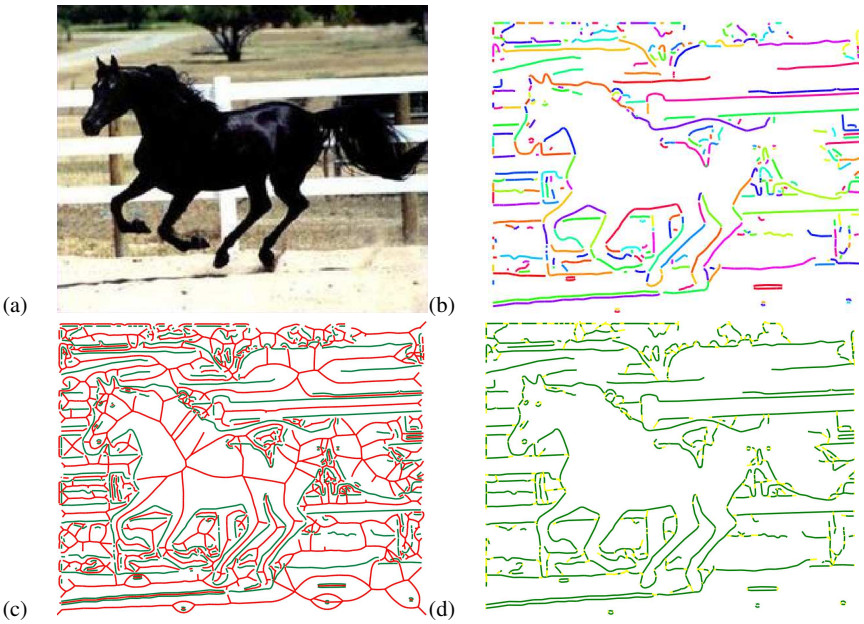


Figure 3: The same steps in Figure 2 are shown for another horse image.

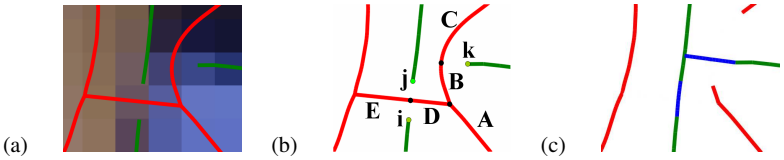
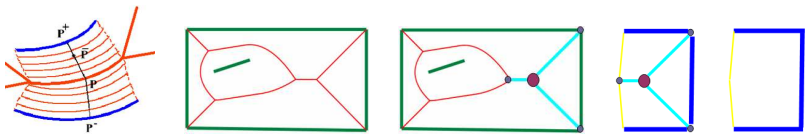


Figure 4: (a) Image curves shown in green and shock graph in red (b) D, E and A are degenerate edges, suggesting the closure of (i-j), (j-i) and (i-k) respectively. B and C are semi-degenerate edges suggesting to form a T-junction from k to the contour. (b) Completion curves in blue after the gap (i-j) is closed and a T-junction is formed based on the closure criteria.

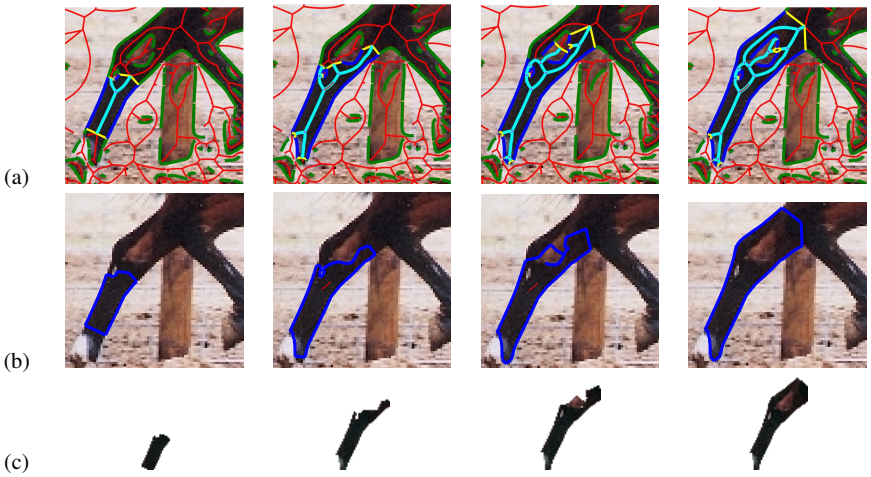
boundary of this region is partially detected curve fragments shown in blue in Figure 5d, and partially by virtual contours imposed by end-nodes, shown in yellow. Figure 6a shows four subgraphs of increasing depths for a selected node on a real image example. Observe that when the subgraph contains a loop, *e.g.* due to a spurious edge, the fragment boundary does not contain this inner boundary, effectively removing it from consideration.

The shock patch fragments then consist of an outer contour as well as an appearance of the inner region. Each shock graph node produces shock patch fragments at all depths 1,2, *etc*. This collection of shock patch fragments is highly redundant, since adjacent nodes produce very similar fragments and since fragments from the same node but at different depths are similar. Furthermore, low-depth patches are often not very informative. Therefore, we subsample depths $(d_1, d_2, \ldots, d_n)$ = (6, 9, 12, 15, 18), and use the extent of overlap to remove similar patches generated by nearby nodes. All patches with 80% or more overlap are considered equivalent, and represented by the patch with the highest appearance contrast. This reduces the number of fragments from thousands to about 30-100



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

Figure 5: (a) Visual fragment (b) A simple shape with boundaries in green and shock graph in red (c) A subgraph at depth 1 (d) Induced boundaries in blue, virtual boundaries in yellow (e) Shock patch fragment.



(a)

(b)

(c)

Figure 6: (a) Shock subgraphs at depths 1, 2, 3 and 4, respectively. The shock graph is shown in red and the subgraph in light green, image boundaries are shown in green, shock patch boundaries in blue. (b) shows the simple closed boundary in blue traced from the outer face of the subgraph. (c) Four shock patch fragments.

per image, as shown for the horse examples of Figures 2 and 3 in Figure 7.

4 Object Matching and Detection using Shock Patches

Shock patch fragments can depict either object fragments, effectively implementing a partial figure-ground segmentation, they can be pieces of the background, or object combined with the background, for example in Figure 7 some fragments depict meaningful object parts, *e.g.*, the horse head, limb, torso, *etc.*, while others do not clearly map to a distinguishable part. When we compare the two horse images, we do not expect any similarities between the second type of fragments, while we do expect some similarity between the head, limb, torso *etc.* between the two sets, and this can be confirmed in Figure 8.

Our approach therefore relies on finding similar fragments between the two sets. Fragment similarity can be measured by comparing the shape and appearance of the two fragments. As tempting as it is, we have excluded appearance from our current fragment similarity measure, both to explore the limits of a shape-based measure, and also because fragment appearance similarity is very well explored elsewhere in the patch-based object

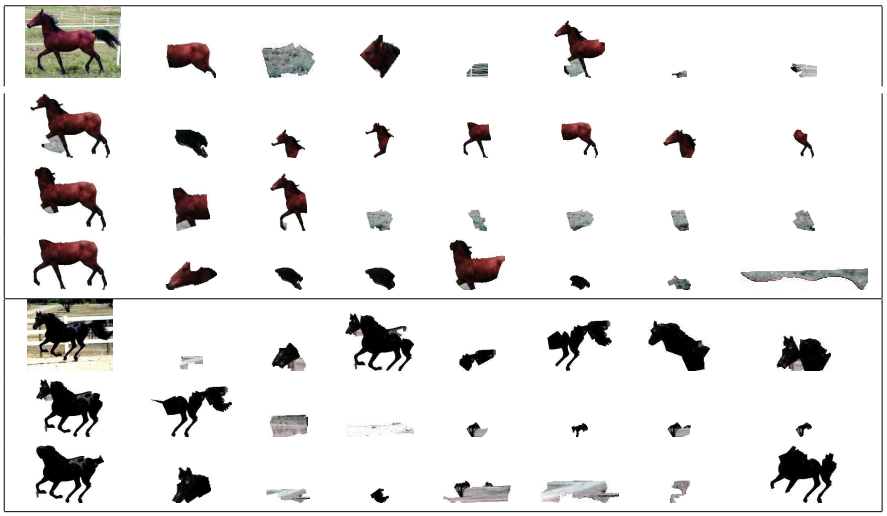


Figure 7: Example patches with depths (6, 9, 12, 15, 18) from the example horses. Observe that all major body parts are covered.

recognition work in the form of local descriptors [14]. We expect that the addition of appearance would improve our recognition results.

Fragment shape similarity should be measured using an algorithm that is robust against occlusions. This is because one can view the fragments as partial occlusions of the figure, *i.e.*, when a horse's torso is compared to a horse. In addition, it must capture intra-class shape variations very well. We therefore use the approach proposed by Sebastian *et al.* [23] which uses an edit-distance algorithm based on shock transitions [8] which handles both well. Figure 8 shows some example matches and non-matches.

5 Results

We propose an object detection and classification algorithm using only a few segmented object images as the training set (one in the case of this paper). We tested our system on the Horse-side class used in [18] consisting of 88 horse images and 88 background images. First, to explore the strength of the matching algorithm we used a single shock patch obtained from the silhouette mask of one of the images as the training set, shown in the second row of Figure 8. We matched all the test image patches to our model patch and declare a detection if top 3 matches are below a given similarity threshold and return the detection box to be the union of top 3 image patches. See Figure 8 for the top 5 matches of some test images. An object is deemed correctly detected if the overlap of the bounding boxes (detection vs ground truth) is greater than 50%. Our recall with best threshold settings is 75% with a precision of 85%. There are two reasons for the low recall rate, one is the use of a single model leading to large deviation, *e.g.* as the pose varies, and the other is that the partial matching of a single fragment to a full model degrades as the ratio of fragment area to the full model decreases. Observe from Figure 8 bottom row that the head of the example horse is correctly matching to the head of the training image

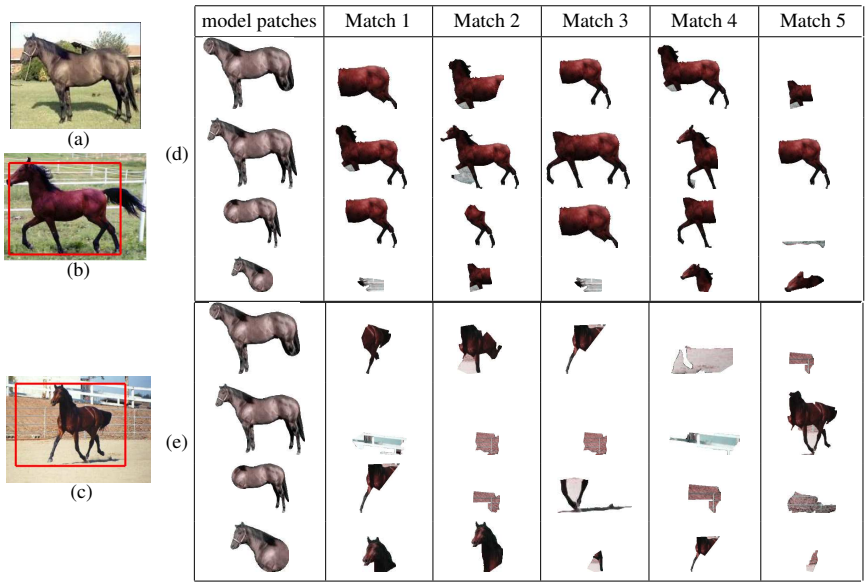


Figure 8: This figure illustrates the similarity between the model horse (a) fragments to fragments from two other horse images (b, c), as measured by the shock graph edit distance [23]. The detection boxes outputted by our system are superimposed on the test images shown in (b, c).

despite the pose difference, but there is not sufficient shape content to match against a full model. This motivates replacing the model by the model shock patch fragments. With this modification and the constraint that at least two model patches' top 3 matches should be within the similarity threshold, recall rate at the best threshold settings increases to 92%, with 85% precision. This second test image is classified correctly in this setting.

In conclusion, we have presented a shape based object detection and classification system which does not require involved training/learning stages and which has promising results on a difficult test set. These results can be improved by making use of spatial constraints which are naturally imposed by the shock topology of the training image, *e.g.*, head patch should be detected in correct relative position and orientation with respect to the torso patch *e.t.c*. Technical enhancements such as the implementation of the loop transform to further clean the curve fragment set, the inclusion of a few more training examples, and the use of appearance in the fragment similarity computation, all should lead to improvements in the recognition rate. Our main contribution in this paper is to present a novel method to generate fragments of images and illustrate their use in a generic object recognition and detection task.

Acknowledgements: This material is based on work supported by the National Science Foundation under Grant No. 0413215.

References

[1] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Trans. on Pattern Anal. Mach. Intell.*, 20(11):1475–1490, 2004.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):509–522, 2002.

[3] D. Crandall, P. Felzenszwalb, and D. Huttenlocher. Spatial priors for part-based recognition using statistical models. In *CVPR*, San Diego, CA, 2005.

[4] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *PAMI*, 28(4):594–611, 2006.

[5] L. Fei-Fei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In *CVPR*, pages 524–531, 2005.

[6] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, Madison, WI, 2003.

[7] R. Fergus, P. Perona, and A. Zisserman. A visual category filter for google images. In *ECCV*, 2004.

[8] P. J. Giblin and B. B. Kimia. On the local form and transitions of symmetry sets, medial axes, and shocks. *IJCV*, 54(Issue 1-3):143–157, August 2003.

[9] F. Jurie and C. Schmid. Scale-invariant shape features for recognition of object categories. In *CVPR*, 2004.

[10] M. F. Kelly and M. D. Levine. Annular symmetry operators: A method for locating and describing objects. In *ICCV*, 1995.

[11] B. B. Kimia, A. R. Tannenbaum, and S. W. Zucker. Shapes, shocks, and deformations, I: The components of shape and the reaction-diffusion space. *IJCV*, 15(3):189–224, 1995.

[12] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Extending pictorial structures for object recognition. In *BMVC*, pages 789–798, 2004.

[13] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[14] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(10):1615–1630, 2005.

[15] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. J. V. Gool. A comparison of affine region detectors. *IJCV*, 65(1-2):43–72, 2005.

[16] R. C. Nelson and A. Selinger. A cubist approach to object recognition. In *ICCV*, pages 614–621, 1998.

[17] A. Opelt, A. Pinz, and A. Zisserman. A boundary-fragment-model for object detection. In *ECCV*, 2006.

[18] A. Opelt, A. Pinz, and A. Zisserman. Incremental learning of object detectors using a visual shape alphabet. In *CVPR*, 2006.

[19] O. C. Ozcanli, A. Tamrakar, B. B. Kimia, and J. L. Mundy. Augmenting shape with appearance in vehicle category recognition. In *CVPR*, 2006.

[20] M. A. Ruzon and C. Tomasi. Edge, junction, and corner detection using color distributions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1281–1295, 2001.

[21] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or "how do i organize my holiday snaps?". In *ECCV*, volume 1, pages 414–431, 2002.

[22] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE PAMI*, 19(5):530–535, 1997.

[23] T. Sebastian, P. Klein, and B. Kimia. Recognition of shapes by editing their shock graphs. *PAMI*, 26:551–571, May 2004.

[24] A. Selinger and R. C. Nelson. A perceptual grouping hierarchy for appearance-based 3d object recognition. *Computer Vision and Image Understanding*, 76(1):83–92, 1999.

[25] J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. In *ICCV*, 2005.

[26] K. Siddiqi and B. B. Kimia. A shock grammar for recognition. In *Proc. CVPR*, pages 507–513, 1996.

[27] K. Siddiqi, A. Shokoufandeh, S. J. Shokoufandeh, and S. W. Zucker. Shock graphs and shape matching. In *ICCV*, pages 222–229, 1998.

[28] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering objects and their localization in images. In *ICCV*, pages 370–377, 2005.

[29] E. B. Sudderth, A. B. Torralba, W. T. Freeman, and A. S. Willsky. Learning hierarchical models of scenes, objects, and parts. In *ICCV*, pages 1331–1338, 2005.

[30] A. Tamrakar and B. B. Kimia. Medial visual fragments as an intermediate image representation for segmentation and perceptual grouping. In *Proc. of POCV*, page 47, 2004.

[31] A. Tamrakar and B. B. Kimia. No grouping left behind: From edges to curve fragments. In *sub. to ICCV*, 2007.

[32] H. Tek and B. B. Kimia. Symmetry maps of free-form curve segments via wave propagation. In *ICCV*, 1999.

[33] D. Tell and S. Carlsson. Combining appearance and topology for wide baseline matching. In *ECCV*, 2002.

[34] M. Vidal-Naquet and S. Ullman. Object recognition with informative features and linear classification. In *ICCV*, pages 281–288, Nice, France, 2003.

[35] G. Wang, Y. Zhang, and L. Fei-Fei. Using dependent regions for object categorization in a generative framework. In *CVPR*, volume 2, pages 1597–1604, 2006.

Implicit Active Model using Radial Basis Function Interpolated Level Sets

Xianghua Xie and Majid Mirmehdi
Department of Computer Science
University of Bristol, Bristol BS8 1UB, England.
{xie,majid}@cs.bris.ac.uk

Abstract

Building on recent work by others that introduced RBFs into level sets for structural topology optimisation, we introduce the concept into active models and present a new level set formulation able to handle more complex topological changes, in particular perturbation *away from the evolving front*. This allows the initial contour or surface to be placed arbitrarily in the image. The proposed level set updating scheme is efficient and does not suffer from self-flattening while evolving which will cause large numerical error. Unlike conventional level set based active models, periodic re-initialisation is also no longer necessary and the computational grid can be much coarser, thus, it has great potential in modelling in high dimensional space. We show results on synthetic and real data for active modelling in 2D and 3D.

1 Introduction

The application of the level set method [7] to the active contour model has enabled the latter to adapt to complex topologies. It avoids the need to reparameterise the curve and the contours are able to split or merge in order to capture an unknown number of objects. However, the original level set based active contour [1] proved to be of limited use in real applications as it assumes that contours reach the object boundaries at roughly the same time. Thus, it often suffers from weak edge leakage. The development of improved external forces, in particular region based methods, such as [10, 8], have greatly improved the performance of level set based snakes. They are generally less initialisation dependent and exhibit better ability in handling textures and image noise interference. Among many others, some realised, practical applications can be found in [10, 9].

The extension of the active contour model into the active surface model is relatively straightforward due to their implicit representation in the level set scheme. However, this implicit representation embeds the contour or surface into a higher dimensional space which needs to be updated iteratively as a whole, becoming more computationally expensive than traditional parametric approaches. The evolution of the embedded contour or surface is solved using partial differential equations (PDEs) which in most cases involves costly finite difference methods (FDM).

More importantly, in conventional level set methods, active contours or surfaces are not able to create topological changes *away from the zero level set* where the deformable

contours or surfaces are embedded [9, 11]. This means, for example, that the level sets would miss holes inside objects. In order to accurately solve the associated PDEs using FDM, a local method, it requires the implicit function to be smooth and maintained to be so while evolving. Thus, re-initialisation is usually necessary in order to achieve numerical stability. Although alternative methods without re-initialisation are available, they often require dedicated extension of the speed function defined on the contour.

As a primary interpolation tool, radial basis functions (RBFs) have received increasing attention in solving PDE systems in recent years. For example, Cecil *et al.* [2] used RBFs to generalise conventional FDM on a non-uniform (unstructured) computational grid to solve the high dimensional Hamilton-Jacobi PDEs with high accuracy. Very recently, Wang *et al.* [11] interpolated level set functions using RBFs and transformed the Hamilton-Jacobi PDEs into a system of ordinary differential equations (ODEs) for structural topology optimisation in 2D.

In this paper, we adapt the approach presented for structure design in [11] to apply to active modelling and show how our proposed model greatly enhances the performance of active models. Following [11], we interpolate the initial level set function using RBFs and treat the implicit contour/surface propagation as an ODE problem, which is much easier and more efficient to solve. However, the updating scheme proposed in [11] was found to be unsuitable for active modelling. A simple yet effective normalisation scheme is proposed to resolve this issue. This new active model exhibits significant improvements in initialisation invariancy, convergence ability, and topology adaptability. The initial contour or surface is embedded into an implicit function derived from the distance transform in the way same as the conventional level set approach. However, we then interpolate it using RBFs which can be placed on a much coarser grid. The interpolation is characterised by its expansion coefficients. Thus, deforming the original implicit function is achieved by updating the expansion coefficients. Re-initialisation is found no longer necessary and perturbation away from the zero level is allowed to obtain more sophisticated topological changes. The contour or surface can therefore be initialised anywhere in the image. We show an implementation of this RBF level set method in a region based active contour model. The extension of this method to 3D on synthetic data is also demonstrated.

Notably, very recently in [5], Morse *et al.* placed RBFs at contour landmarks to implicitly represent the active contour, thereby avoiding the manipulation of a higher dimensional function. However, the method requires dynamic insertion and deletion of landmarks which is non-trivial. Similar to the parametric representation, the resolution and position of the landmarks can affect the accuracy of contour representation.

In the next section we present a brief review of the conventional level set method, RBF interpolation, the proposed RBF level set evolution, and its application to a region based active contour model. The extension to 3D is presented in Section 3. Conclusions and future work are discussed in Section 4.

2 Proposed Method

2.1 Level Set Representation

Using level sets [7], a contour or surface is implicitly represented by a multi-dimensional scalar function with the moving front embedded at the zero level set. Let C and Φ denote the moving front and the level set function respectively. The relationship between these two can be expressed as: $C = \{\mathbf{x}|\Phi(\mathbf{x}) = 0\}$ where $\mathbf{x} \in \mathbb{R}^n$, and subject to $\Phi(\mathbf{x}) > 0$ for

$\mathbf{x}$ inside the front and $\Phi(\mathbf{x}) < 0$ for $\mathbf{x}$ outside. This representation is parameter free and intrinsic. Considering the front (contour or surface) evolving according to $dC/dt = F\mathcal{N}$ for a given function F (where $\mathcal{N}$ denotes the inward unit normal), then the embedding function should deform according to $\partial\Phi/\partial t = F|\nabla\Phi|$, where F is computed on the level sets. By embedding the evolution of C in that of Φ, topological changes of C, such as split and merge, are handled automatically.

The level set function is commonly initialised using the signed distance transform and its evolution numerically solved using FDM with the upwind scheme [7]. The numerical error using this local approximation method may gradually accumulate and can contaminate the solution. Thus, periodic re-initialisation of the level set function is usually applied to maintain numerical stability. The conventional level set method generally prevents topological changes taking place away from the developing front which restricts other forms of topological changes, such as developing holes inside objects. The method presented here will allow the level set contour or surface to deal with regions away from the evolving front by initiating new fronts in the level set and thus capture holes or inner boundaries of objects. This makes the active contour or surface framework not only much more successful but also initialisation invariant.

2.2 RBF Interpolated Level Set Function

Similar to recent works by Cecil *et al.* [2] and Wang *et al.* [11], we interpolate the level set function $\Phi(\mathbf{x})$ using a certain number of RBFs. Each RBF, ψ_i, is a radially symmetric function centred at position $\mathbf{x}_i$. Only a single function ψ is used to form this family of RBFs. The multiquadric spline, found to be one of the best for RBF interpolation [3] is used here, with the RBFs then written as:

$$\psi_i(\mathbf{x}) = \psi(||\mathbf{x} - \mathbf{x}_i||) = \sqrt{(\mathbf{x} - \mathbf{x}_i)^2 + c_i^2}, \tag{1}$$

where c_i is usually treated as a constant for all RBFs. The interpolation is expressed as:

$$\Phi(\mathbf{x}) = p(\mathbf{x}) + \sum_{i=1}^{N} \alpha_i \psi_i(\mathbf{x}), \tag{2}$$

where N denotes the number of RBFs, α_i are the expansion coefficients of the corresponding RBF, and $p(\mathbf{x})$ is a first-degree polynomial, which in the 2D case can be written as $p(\mathbf{x}) = p_0 + p_1 x + p_2 y$.[1] To ensure a unique solution to this RBF interpolation, the expansion coefficients must satisfy $\sum_{i=1}^{N} \alpha_i = \sum_{i=1}^{N} \alpha_i x_i = \sum_{i=1}^{N} \alpha_i y_i = 0$. These N number of RBFs are uniformly distributed in the domain and their centre values, denoted by $f_1, ..., f_N$, are given by the level set function. The RBF interpolant then can be obtained by solving the following linear system:

$$\mathbf{H}\boldsymbol{\alpha} = \mathbf{f}, \quad \text{where} \quad \mathbf{H} = \begin{pmatrix} \mathbf{A} & \mathbf{P} \\ \mathbf{P}^T & 0 \end{pmatrix}, \tag{3}$$

$$\boldsymbol{\alpha} = [\alpha_1 \ \cdots \ \alpha_N \ p_0 \ p_1 \ p_2]^T, \qquad \mathbf{f} = [f_1 \ \cdots \ f_N \ 0 \ 0 \ 0]^T,$$

and $\mathbf{A}_{i,j} = \psi_j(\mathbf{x}_i), i, j = 1, \ldots, N$, $\mathbf{P}_{i,j} = p_j(\mathbf{x}_i), i = 1, \ldots, N, j = 1, 2, 3$, and p_j are the basis for the polynomial. Thus, the RBF interpolation of the level set function in (2)

[1] For simplicity, we present the solution in 2D. Its solution in higher dimension is straightforward.

can be written as:

$$\Phi(\mathbf{x}) = \boldsymbol{\Psi}^T(\mathbf{x})\boldsymbol{\alpha}, \tag{4}$$

where $\boldsymbol{\Psi}(\mathbf{x}) = [\psi_1(\mathbf{x}) \ \cdots \ \psi_N(\mathbf{x}) \ 1 \ x \ y]^T$.

2.3 Active Modelling using RBF Level Set

As stated in Section 2.1, the deformation of the active contour is achieved by propagating the level sets along their normal directions according to a localised speed which is usually image dependent. It can be expressed as the following PDE:

$$\frac{\partial \Phi}{\partial t} + F|\nabla \Phi| = 0, \tag{5}$$

where F is the speed function along the normal direction. Unlike the conventional level set method, here we have a level set function interpolated by RBFs. Following [11], we assume that time and space are separable and the time dependence of the level set function is now due to the RBF interpolation, i.e. the expansion coefficients. Updating the level set function is now considered as updating the RBF expansion coefficients. In other words, the expansion coefficients become time dependent: $\Phi = \boldsymbol{\Psi}^T(\mathbf{x})\boldsymbol{\alpha}(t)$. Thus, the level set updating function (5) can be re-written as:

$$\frac{\partial \Phi}{\partial t} + F|\nabla \Phi| = \boldsymbol{\Psi}^T \frac{d\boldsymbol{\alpha}}{dt} + F|(\nabla \boldsymbol{\Psi})^T \boldsymbol{\alpha}| = 0. \tag{6}$$

This indicates the original PDE problem can now be treated as an ODE problem. The spatial derivative $\nabla \boldsymbol{\Psi}$ can be solved analytically using the first order Euler's method, also adopted in [11]. Substituting (3) into (6) we have,

$$\mathbf{H}\frac{d\boldsymbol{\alpha}}{dt} + \mathbf{B}(\boldsymbol{\alpha}) = 0, \tag{7}$$

where

$$\mathbf{B}(\boldsymbol{\alpha}) = [F(\mathbf{x}_1)|(\nabla \boldsymbol{\Psi}^T(\mathbf{x}_1))\boldsymbol{\alpha}| \ \ \ldots \ \ F(\mathbf{x}_N)|(\nabla \boldsymbol{\Psi}^T(\mathbf{x}_N))\boldsymbol{\alpha}| \ \ 0 \ \ 0 \ \ 0]^T \tag{8}$$

The solution can be obtained by iteratively updating the expansion coefficients:

$$\boldsymbol{\alpha}(t^{n+1}) = \boldsymbol{\alpha}(t^n) - \Delta t \mathbf{H}^{-1}\mathbf{B}(\boldsymbol{\alpha}(t^n)). \tag{9}$$

The updating of the level set function starts from interpolating its initial state using RBFs. As usual, the initial level set function is obtained from the signed distance transform. Then RBFs are uniformly spread across the domain and the interpolation takes place which gives us the initial value of the expansion coefficients, $\boldsymbol{\alpha}$. The interpolated level set then is evolved according to (9) and (2). Unlike conventional level set approaches where the upwind scheme [7] is often used and re-initialisation is applied to maintain numerical stability, the coefficient updating is much simpler and efficient and does not require re-initialisation.

Although (9) has been proven useful in structure optimisation in [11], a direct application of this updating scheme was found to be unsuitable for active contour models. An example is given in Fig. 1 where a circular shape is embedded in an initial level set function. A constant force is applied to this active contour, i.e. F is a constant. This force

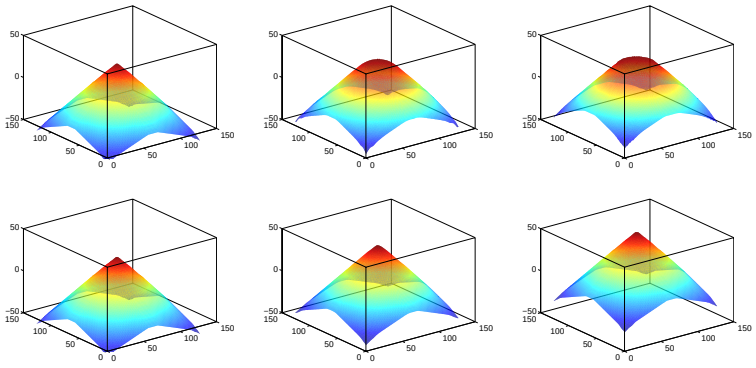


Figure 1: Updating RBF level set using non-normalised and normalised schemes - first row: Non-normalised scheme; second row: proposed normalised scheme.

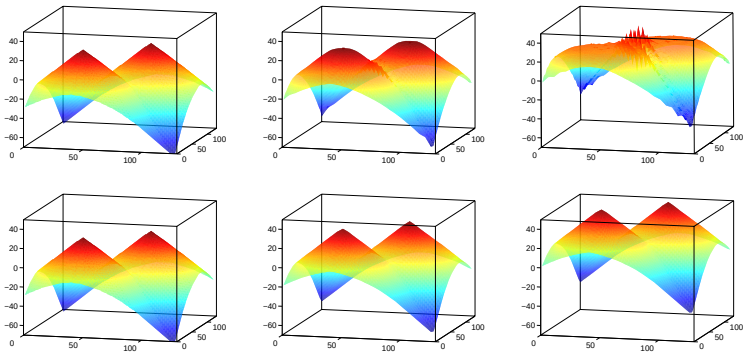


Figure 2: Updating the RBF level set using non-normalised and normalised schemes - first row: Non-normalised scheme; second row: Proposed normalised scheme.

expands the contour outwards which should generally lift the level set function. However, as shown in the first row, the top of the level set function becomes stationary and gradually turns into a flat surface. This is due to the gradient values of the RBF interpolated level set at those points being close to zero ($|(\nabla \boldsymbol{\Psi}^T(\mathbf{x}_i))\boldsymbol{\alpha}| \rightarrow 0$) and based on (8) and (9) the expansion coefficients at those places would evolve much slower. As a result, the level set function tends to get flattened and this is undesirable when topological changes should be taking place. See for example in Fig. 2 where two circles are expanding due to the same constant force. The valley in the level set function is affected and introduces numerical artifacts, and finally contaminates the solution as shown in the last image in the first row, indicated by the highly irregular spikes in the level set function. Special care is thus necessary, for example using dedicated velocity extension.

Fortunately, in active contours the direction of the speed along the normal has dominant effect on the final segmentation, not its magnitude. Since the gradient of the level

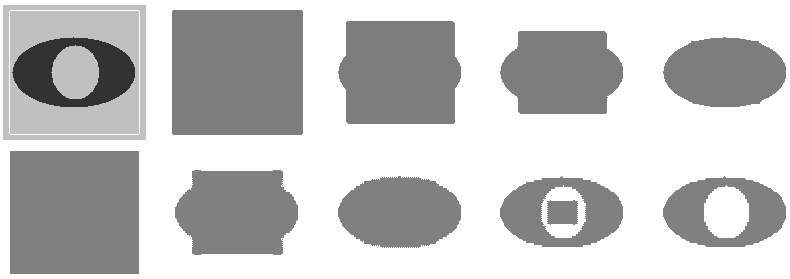


Figure 3: More complex topological changes are readily achievable - first row: initial snake and recovered shape using conventional level set method; second row: recovered shape using proposed method. The final images in both rows show the stabilised results.

set function is generally smoothly varying, a simple yet highly effective solution can be devised to solve this problem. We modify the speed function by "normalising" it against the local gradient estimated from the RBF interpolants, i.e.

$$F'(\mathbf{x}_i) = \frac{1}{|(\nabla \mathbf{\Psi}^T(\mathbf{x}_i))\boldsymbol{\alpha}|} F(\mathbf{x}_i). \tag{10}$$

Note that due to this global modelling using RBFs, the gradient is dependent on all the RBF centres across the domain, instead of local neighbours. Thus, the gradient near the advancing front is unlikely to be zero, i.e. this normalisation will be unlikely to disturb the developing front. Eq. (8) then simplifies to:

$$\mathbf{B}(\boldsymbol{\alpha}) = [F(\mathbf{x}_1) \ \ \ldots \ \ F(\mathbf{x}_N) \ \ 0 \ \ 0 \ \ 0]^T. \tag{11}$$

Updating the expansion coefficients and hence the level set are now even simpler and more efficient. The second row in Fig. 1 shows the results using the normalised approach. The level set function does not get flattened while updating the expansion coefficients. Topological changes, for example merging shown in the second row in Fig. 2, can be conveniently handled, in contrast to the non-normalised scheme (shown in the first row).

One of the main advantages of using a RBF interpolated level set to represent an active contour is that more sophisticated topological changes, besides merging and splitting, can be readily achieved. Let F be a region indication function, i.e. $F < 0$ for points inside an object of interest and $F > 0$ for the rest. In Fig. 3, the object of interest is shown in dark gray, and the initial snake is drawn in white. The snake using the conventional level set scheme with re-initialisation failed to recover the hole in the object as periodic re-initialisation prevents it from doing so. The proposed RBF based level set method successfully recovered the shape without dedicated effort in monitoring the front propagation. This occurs because the proposed method uses RBF interpolants to estimate the level set gradient, a global estimation instead of a local one. Front propagation is then unlikely to introduce oscillation around the zero level set. Thus re-initialisation is not necessary to maintain stability. The proposed RBF expansion coefficients updating scheme prevents other level sets, away from the evolving front, from flattening themselves so that these level sets are sensitive enough to sufficient gradient changes for the RBF interpolated front to grow new fronts (i.e contours or surfaces).

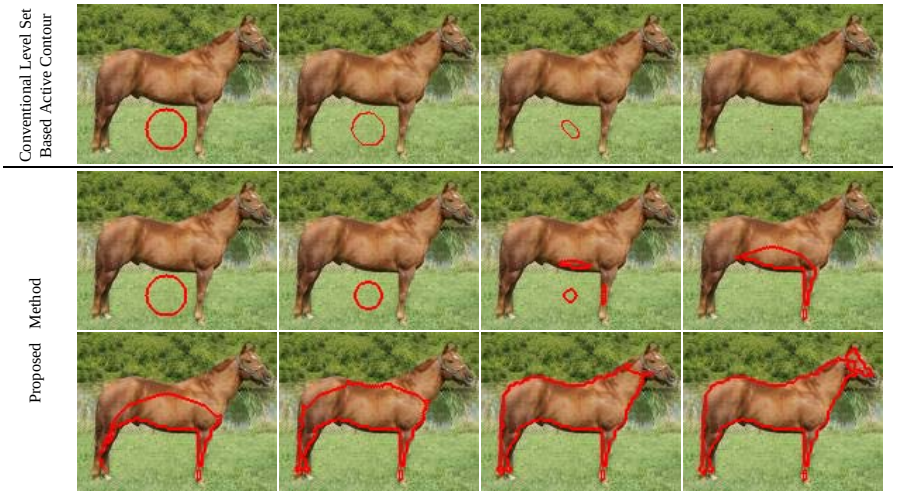


Figure 4: Comparative result on real image - first row: Segmentation result using conventional level set with the initial snake forced to shrink; second and third rows: Results using proposed RBF level set method.

2.4 A Region Based Active Contour Model using RBF Level Sets

We now present a region based active contour model as a demonstration of the proposed RBF level set method. As mentioned earlier in Section 1, region based methods generally perform better in the presence of weak edges and image noise interference. More importantly in relevance to this work, region based methods are considered much less initialisation dependent. There are two classes of popular region based approaches. One is based on the well-known Mumford-Shah formulation [6], where the contours compete with each other while preserving the piecewise constant assumption. The other, such as the works in [10, 8], globally model the image data and the active contour evolves to maximise its posterior. We opt for the second approach and model the image data using Gaussian Mixture Models (GMM). Note we are not advocating a region based approach or this particular GMM based method, but we employ these to demonstrate the performance of our proposed RBF level set method. Our aim is to give a comparative study of the proposed RBF level set method with the conventional level set approach *in the same active contour framework*.

The colour (or intensity) histogram of a given image is modelled using GMM. Each pixel is then assigned posterior probabilities for each class. Let u denote the posterior probability of the class of interest. The GMM region based active contour can be formulated as:

$$\frac{dC}{dt} = (1 - \frac{1}{m})u\mathcal{N}, \tag{12}$$

where m is the number of classes and $\frac{1}{m}$ is the average expectation of a class probability. Its level set representation takes the following form:

$$\frac{\partial \Phi}{\partial t} = (1 - \frac{1}{m})u|\nabla \Phi|. \tag{13}$$

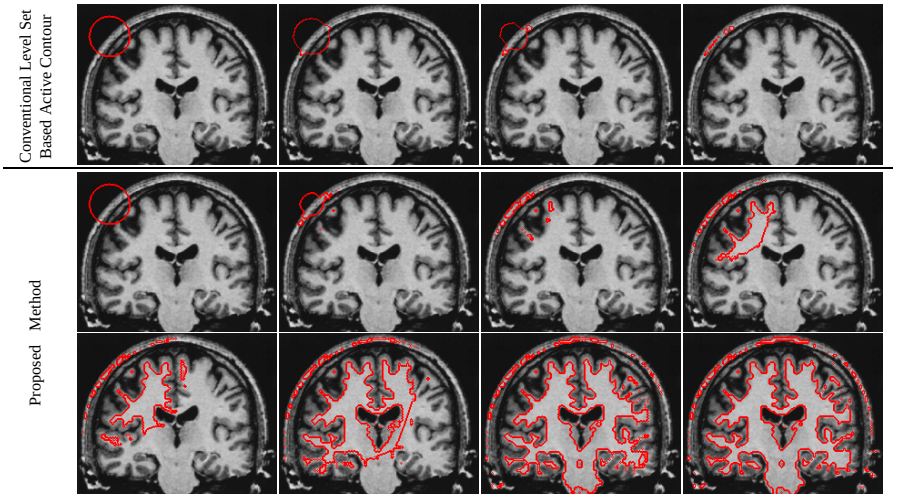


Figure 5: Comparative result on real image - first row: Segmentation result using conventional level set; second and third rows: Results using proposed RBF level set method.

For simplicity we ignore the internal contour regularisation terms, but use the image dependent force term alone to deform the active contour. The contour is supposed to expand and shrink to maximise the posterior of the regions of interest. With the proposed method, new contours can even grow out in regions away from existing contours, which is not possible for the conventional level set approach. Equally importantly, the initial contour can be forced to vanish from the image domain while newly appearing fronts are able to localise the regions. This gives significant improvement in initialisation invariancy and achieves global minimum, instead of local minimum (as demonstrated earlier in Fig. 3).

Fig. 4 shows the comparative results of the GMM region snake using the conventional level set approach (top row) and the proposed RBF level set method (rows 2 and 3). The initial snake was placed outside the object of interest and was forced to *shrink*. The conventional method failed to localise the object while the proposed method succeeded by growing out new contours inside the object. In this case, the conventional method requires the initial snake to be specifically placed overlapping or inside the object. Another example is given in Fig. 5, where multiple regions exist. The proposed method could localise all the regions that were indicated by the function u, while conventional level sets could only capture those that the initial contour had touched.

3 Extension to 3D

Similar to the conventional level set method, the extension of the proposed method to higher dimensions is straightforward. Even better, the proposed method demands only a much coarser mesh grid. The RBF centres can be more loosely placed in 3D, instead of the full pixel grid often used in conventional level set approaches. Also, solving the ODE system in 3D is much easier than solving the PDE system. The updating of the expansion coefficients are efficient and again does not require re-initialisation of the level set function. The main computation cost comes from interpolating the initial level set and

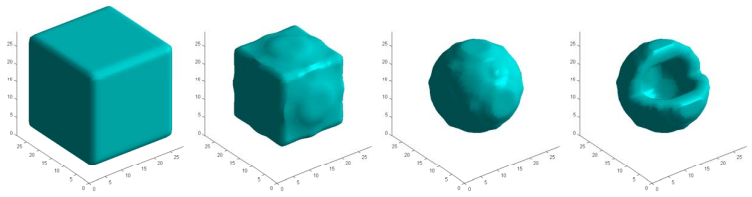


Figure 6: Recovering a hollow sphere using proposed method - from left: Initial deformable surface, evolving deformable surface, stabilised surface, and the stabilised surface with a section cut away to show the hole captured inside.

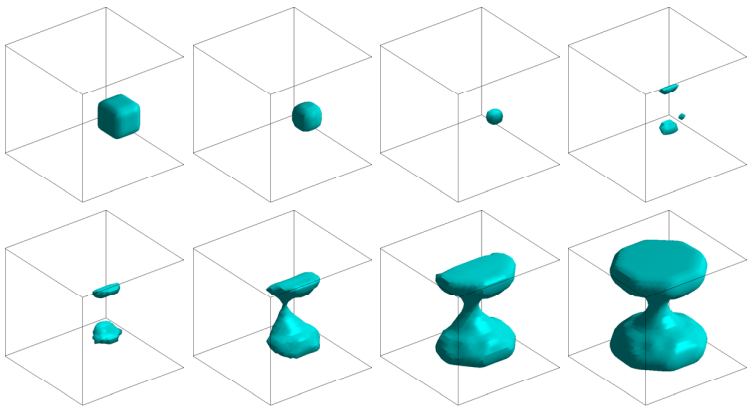


Figure 7: Arbitrary initialisation - The initial surface is placed outside the object and is forced to shrink, but the proposed method allows the level set to deform further to develop a zero level set outside the initial surface and recover the object.

reconstructing the level set function after it stabilises. However, there are several methods available to speed up the process, such as the Fast Multipole Method (FMM) [4].

We examine the ability of the proposed method in handling complex 3D topologies and initialisation invariancy. We apply the 3D RBF level set method on synthetic data and evolve the active surface according to (5), where $F < 0$ for regions inside the 3D objects and $F > 0$ otherwise, as before. In Fig. 6, the target object was a hollow sphere. The initial surface was placed to surround the object and was forced to shrink to capture the object boundaries. With the proposed RBF level set method, not only was the outer boundary localised, but also the boundary inside was captured, i.e. as the active surface was deforming, a new zero level set developed inside the object. The next example given in Fig. 7 shows that the region indication function shrinks the active surface that initialised outside the target object. There was no intersection between the initial surface and the object, neither when the initial surface deformed and disappeared. However, the proposed method allows the level set to deform further to "grow" outside the initial surface and finally recovers the object. This again demonstrates the method's initialisation independence feature. In the final example shown in Fig. 8, we demonstrate the ability of the proposed method in modelling very complex geometry in 3D.

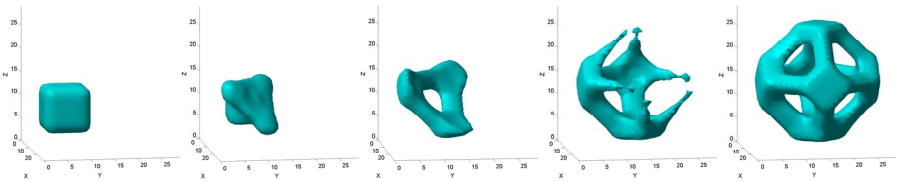


Figure 8: Recovering a complex 3D shape.

4 Conclusion

We have presented a novel method to perform implicit modelling using RBFs. The proposed method has a number of advantages over the conventional level set scheme: (a) The evolution of the level set function is considered as an ODE problem rather than a much more difficult PDE problem; (b) Re-initialisation of the level set function was found no longer necessary for this application; (c) More complex topological changes, such as holes within objects, are comfortably found; (d) The active contour and surface models using this technique are initialisation independent; (e) The computational grid can be much coarser, hence it is more computationally cheaper when updating the level set function, particularly in high dimensional spaces. Future work includes implementing a fast implementation of RBF fitting and reconstruction, and applying this method to large scale 3D segmentation problems.

References

[1] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contour. *IJCV*, 22(1):61–79, 1997.

[2] T. Cecil, J. Qian, and S. Osher. Numerical methods for higher dimensional Hamilton-Jacobi equations using radial basis functions. *Journal of Computational Physics*, 196:327–347, 2004.

[3] R. Franke. Scattered data interpolation: Tests of some methods. *Mathematics of Computation*, 38:181–200, 1982.

[4] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73:325–348, 1987.

[5] B. Morse, W. Liu, T. Yoo, and K. Subramanian. Active contours using a constraint-based implicit representation. In *CVPR*, pages 285–292, 2005.

[6] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Commun. Pure Appl. Math.*, 42(5):577–685, 1989.

[7] S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[8] N. Paragios and R. Deriche. Geodesic active regions and level set methods for supervised texture segmentation. *IJCV*, 46(3):223–247, 2002.

[9] R. Ramlau and W. Ring. A Mumford-Shah level-set approach for the invesion and segmentation of X-ray tomography data. *Journal of Computational Physics*, 221:539–557, 2007.

[10] J. Suri. Two-dimensional fast magnetic resonance brain segmentation. *IEEE Engineering in Medcine and Biology*, 20(4):84–95, 2001.

[11] S. Wang, K. Lim, B. Khoo, and M. Wang. An extended level set method for shape and topology optimization. *Journal of Computational Physics*, 221:395–421, 2007.

Using Priors for Improving Generalization in Non-Rigid Structure-from-Motion

S. I. Olsen[1] **A. Bartoli**[2,1]
[1] DIKU, Copenhagen, Denmark [2] LASMEA, Clermont-Fd, France

Abstract

This paper describes how the generalization ability of methods for non-rigid Structure-from-Motion can be improved by using priors. Most point tracks are often visible only in some of the images; predicting the missing data can be important. Previous Maximum-Likelihood (ML)-approaches on implicit non-rigid Structure-from-Motion generalize badly. Although the estimated model fits well to the visible training data, it often predicts the missing data badly. To improve generalization we propose to add a temporal smoothness prior and a continuous surface shape prior to an ML-approach. The temporal smoothness prior constrains the camera trajectory and the configuration weights to behave smoothly. The surface shape prior constrains consistently close image point tracks to have a similar implicit structure. We propose an algorithm for achieving a Maximum A Posteriori (MAP)-solution and show experimentally that the MAP-solution generalizes far better than the ML-solution. The proposed method is fully automatic: it handles a substantial amount of missing data as well as outlier contaminated data, and automatically estimates the rank of the measurement matrix.

1 Introduction

Non-rigid Structure-from-Motion concerns the simultaneous recovery of the deforming world structure and camera motion from image features. Such analysis extends the classical rigid setup [10] to situations with deforming scenes such as expressive faces, moving cars, *etc*. In [1, 3, 5, 13, 18] methods where the non-rigidity was represented as a linear combination of *basis shapes* were developed and analyzed.

Many previous methods cannot handle situations with missing data [1, 3, 5, 13, 16, 17], but see also [6, 9, 14]. The amount of non-rigidity – the number of basis shapes – often is assumed known [3, 5, 13, 14, 17]. These assumptions seriously limit the applicability of the methods. Recently an implicit low-rank model solving both problems has been proposed [2]. The present paper reviews and extends this approach. One major difference is the use of a MAP-estimation where priors are added to the ML cost function.

Estimating a model from partial data allows one to predict the projection of all world points on all images. The model generalizes well if the predicted points, on frames where the point is not registered, are accurate. In general, the model minimizing the reprojection error – the ML-estimate – does not generalize well. We derive an alternative approach where the optimization function is augmented with a temporal smoothness prior and a surface smoothness prior. The priors we use are different from the ones favoring rigidity in [7, 14].

The proposed MAP-estimator is based on four main steps. First, we compute an initial solution with an existing ML-estimator. Second, we change the implicit coordinate frame such that the temporal smoothness prior is minimized. This ensures that the prior is globally satisfied since we derive a closed-form, optimal solution to this problem. Third, we re-estimate the implicit structure by minimizing a combination of the reprojection error and the surface shape prior. Finally, we jointly refine the motion and structure estimates by nonlinear optimization. Experimental results on simulated and real data show that the generalization ability is greatly improved compared to the standard ML-estimation.

Section 2 reviews the implicit low-rank imaging model, its matching tensors and closure constraints. In section 3 and 4 the rank and model estimation on partial data is described. Sections 5 and 6 describe the proposed priors and their implementation. Section 7 reports the experimental results. Finally, section 8 concludes the paper.

Notation. Vectors are denoted using bold fonts, *e.g.* $\mathbf{x}$ and matrices using sans-serif or calligraphic characters, *e.g.* M or $\mathscr{A}$. Index $i = 1,\ldots,N$ is used for the images, $j = 1,\ldots,M$ for the points. The Hadamard (element-wise) product is written $\odot$. Bars indicate 'centered' data, as in $\bar{\mathsf{X}}$. We use the Singular Value Decomposition, denoted SVD, *e.g.* $\mathsf{X} = \mathsf{U}\Sigma\mathsf{V}^\mathsf{T}$ where U and V are orthonormal matrices, and Σ is diagonal, containing the singular values of X in decreasing order. The operator vect(X) performs column-wise matrix vectorization.

2 The Implicit Low-Rank Non-Rigid Model

The standard rigid model describes the affine projection $\mathbf{x}_{ij}$ of a set of M 3D world points $\mathbf{S}_j$, represented by a $3 \times M$ shape matrix S onto N images represented by a $2N \times 3$ motion matrix J of stacked 2×3 affine camera projection matrices J_i:

$$\mathbf{x}_{ij} = \mathsf{J}_i \mathbf{S}_j + \mathbf{t}_i \tag{1}$$

where $\mathbf{t}_i$ is the position of the i'th camera. The $2N \times M$ matrix X of time varying coordinates is called *the measurement matrix* and has rank $r = 3$.

In the non-rigid case $r > 3$. The low-rank assumption is $r \ll \min\{2N, M\}$. The implicit low-rank non-rigid model extends (1) by letting the camera and shape matrices have dimensions $2N \times r$ and $r \times M$. The model is implicit because no assumptions are made on the replicated block structure of the camera matrices that often is used in explicit approaches *e.g.* [4, 5, 14]. Thus the implicit model is simpler than the explicit one and gives weaker constraints on point tracks. Note that the implicit (basis) shape vectors $\mathbf{S}_j$ are more difficult to interpret in terms of world coordinates. Similarly, the implicit camera matrix J_i (comprising camera pose and configuration weights) does no longer directly relate to the camera orientation.

The factorization of the centered measurement matrix $\bar{\mathsf{X}} = \mathsf{JS} = (\mathsf{J}\mathscr{A})(\mathscr{A}^{-1}\mathsf{S})$ is ambiguous since the equation holds for any full rank $r \times r$ mixing matrix $\mathscr{A}$ defining the coordinate frame in which the cameras and shapes are represented. If X is filled (no missing data), one factorization can be found using SVD as $\bar{\mathsf{X}} = \mathsf{U}\Sigma\mathsf{V}^\mathsf{T}$. The joint implicit camera and shape matrices J and S, are recovered as the r leading columns of *e.g.* U and the rows of $\Sigma\mathsf{V}^\mathsf{T}$ respectively.

Matching tensors [15] relate corresponding points over multiple images. In the non-rigid affine case the matching tensor is a matrix $\mathcal{N}$ whose columns span the d dimensional left nullspace of the centered measurement matrix $\bar{\mathsf{X}}$:

$$\mathcal{N}^{\mathsf{T}}\bar{\mathsf{X}} \quad = \quad 0. \tag{2}$$

The size of $\mathcal{N}$ is $(2N \times d)$ where the tensor dimension is $d = 2N - r$. $\mathcal{N}$ constrains each point track $\bar{\mathbf{x}}_j$ – the j-th column of $\bar{\mathsf{X}}$ – by d linear homogeneous equations $\mathcal{N}^{\mathsf{T}}\bar{\mathbf{x}}_j = \mathbf{0}$. The closure constraints introduced by Triggs in [15] for rigid scenes relate matching tensors to projection matrices. From (1) and (2) and for all implicit shape points $\mathbf{S}_j \in \mathbb{R}^r$ we have $\mathcal{N}^{\mathsf{T}}\mathsf{J}\mathbf{S}_j = \mathbf{0}$, which gives the *$\mathcal{N}$-closure constraint*:

$$\mathcal{N}^{\mathsf{T}}\mathsf{J} \quad = \quad 0. \tag{3}$$

The joint implicit camera matrix J consequently lies in the right nullspace of $\mathcal{N}^{\mathsf{T}}$. From J, $\mathbf{S}_j$ is retrieved point-wise by triangulation. From $\mathbf{x}_j = \mathsf{J}\mathbf{S}_j$ we get $\mathbf{S}_j = \mathsf{J}^{\dagger}\mathbf{x}_j$, where $\mathsf{J}^{\dagger}$ is the pseudoinverse of J. In case of outlier contaminated data the computation of $\mathcal{N}$ as well as the triangulation must be robust so that blunders do not corrupt the computation. We use a RANSAC-based approach called MSAC [12]. Finally, we need r to compute $\mathcal{N}$. As described later we apply the GRIC model selection criterion [11] in conjunction with MSAC to estimate the optimal model size, *i.e.* r.

3 Handling Partial Data

As a number of previous methods [1, 3, 5, 13, 17] we factorize the measurement matrix X using SVD. Since X often is banded because of occlusions and imperfect tracking, handling of missing data is important. As [8, 9] we use a blockwise approach where the measurement matrix is partitioned into a set of highly overlapping blocks. Given r, a d-dimensional matching tensor $\mathcal{N}_b$ can be computed robustly for each block b. For each matching tensor, equation (3) gives a closure constraint on the joint camera matrix J:

$$\begin{pmatrix} 0_{(d\times 2(i_b-1))} & \mathcal{N}_b^{\mathsf{T}} & 0_{(d\times 2(N-i'_b))} \end{pmatrix} \mathsf{J} \quad = \quad 0 \tag{4}$$

where i_b and i'_b are indexes of the first and last frame in block b. Stacking the constraints for all blocks yields an homogeneous linear least squares problem $||\mathsf{A}\mathsf{J}||^2$ which must be solved such that J has full column rank. Without loss of generality the full column rank constraint can be replaced by constraining J to be column orthonormal. A solution is given by the r last columns of V in the SVD $\mathsf{A} = \mathsf{U}\Sigma\mathsf{V}^{\mathsf{T}}$.

For each block the translation vector $\mathbf{t}^b$ is computed prior to $\mathcal{N}_b$. The joint translation vector $\mathbf{t}$ can be found by minimizing the reprojection error $\sum_b \|\mathbf{t}^b - \mathsf{J}_b\mathbf{T}_b - \mathbf{t}_b\|^2$, where $\mathbf{T}$ is the reconstructed centroid, and where the subscript b in J_b, $\mathbf{T}_b$, and $\mathbf{t}_b$ denotes the restriction of the joint matrices and vectors to the frames within block b. The reprojection error is rewritten $\|\mathsf{B}\mathbf{w} - \mathbf{b}\|^2$, where the unknown vector $\mathbf{w}$ contains $\mathbf{T}$ and $\mathbf{t}$. The solution is given by using the pseudo-inverse since there is a r-dimensional ambiguity, making B rank deficient with a left nullspace of dimension r. This correspond to the translational ambiguity between the basis shapes and the joint translation $\mathbf{t}$: $\forall\gamma \in \mathbb{R}^r$, $\mathbf{x}_j = \mathsf{J}\mathbf{S}_j + \mathbf{t} = \mathsf{J}(\mathbf{S}_j - \gamma) + \mathsf{J}\gamma + \mathbf{t} = \mathsf{J}\mathbf{S}'_j + \mathbf{t}'$.

Given the estimates of J and $\mathbf{t}$, the shape vectors S_j could now be computed by a robust minimization of the reprojection error. However, as described in section 6.2, we prefer to postpone this computation until the prior is included.

4 Estimating the rank

With the exception of [1] most of the previous work assumes that the rank of X is given. We propose to use the robust estimator MSAC in conjunction with the GRIC model selection criterion proposed in [11]. Letting k be the number of parameters of the model and $\mathscr{L}$ the log-likelihood of the error distribution obtained by marginalizing a mixture of a Gaussian inlier part and a uniform outlier part, GRIC is defined by: $\text{GRIC} = -2\mathscr{L} + k\log(M)$. Expanding and removing constants the measure becomes:

$$\text{GRIC} = \sum_{j=1}^{M} \rho\left(\frac{e_j^2}{\sigma^2}\right) + Mr\lambda - \frac{1}{2}r(r-1)\log(M) \tag{5}$$

where e_j is the prediction error for the j-th point track, σ^2 is the variance of the point tracker localization error, where $\lambda = 2\log(U) - \log(2\pi\sigma^2)$, and where the function ρ is $\rho(x) = x$ for $x < T$ and $\rho(x) = T$ otherwise. T is the point of intersection of the Gaussian inlier distribution and the uniform outlier distribution and defined by: $T = 2\log\left(\frac{\gamma}{1-\gamma}\right) + (2N-r)\lambda$ where γ is the percentage of inliers. The value of U is determined by the relative weighting of the inlier and outlier distribution and have a major influence on the rank estimation. To estimate U we notice that an alternative approach to the estimation of T is by the value of inverse cumulative χ^2 distribution with $2N-r$ degrees of freedom. For relevant values of $2N-r$ this is approximately linear with a slope of λ. More details are given in [2]. To estimate the rank robustly we must sample the GRIC value repeatedly for all relevant values of r. To limit the computational cost the sequence of trials is divided into groups using gradually narrower intervals of possible rank values.

5 The Priors

Below we motivate and formulate the temporal smoothness prior and the surface shape prior. In the following section the implementation of the priors is described.

5.1 Temporal Smoothness

For most image sequences, the camera motion is smooth. For points on a smoothly deforming surface the configuration weights smoothly vary as well which means that the surface does not 'jump' between poses but rather smoothly interpolates them. Since both the configuration weights and the camera coordinate axes are encapsulated in the J_i-matrices, these should vary smoothly from frame to frame giving the smoothness measure:

$$\mathscr{E}_{\mathsf{J}}(\mathsf{J}) = \sum_{i=1}^{N-1} ||\mathsf{J}_i - \mathsf{J}_{i+1}||^2 = ||\mathsf{L}||^2 \tag{6}$$

where L is the $2(N-1)\times r$ matrix of stacked projection difference matrices. The previously described factorization is ambiguous up to a $r\times r$ full rank mixing matrix $\mathscr{A}$. From (6) we see that $\mathscr{E}_{\mathrm{J}}(\mathsf{J}) \neq \mathscr{E}_{\mathrm{J}}(\mathsf{J}\mathscr{A})$.

5.2 Surface Shape

Points which are close in space also are close in the images. In case of points on a deforming continuous surface the opposite is true as well. Solutions obtained by the method described above does not encourage such behavior. As a consequence the projected trajectories for such close tracks may deviate significantly outside the estimation area. Often the ability to generalize acceptably disappears just 2-5 frames away from the images in which the points are visible. To improve generalization a surface shape prior is imposed. The shape similarity $\alpha(j_1, j_2)$ of two point tracks $j_1 \neq j_2$ is measured by a Gaussian function $e^{\lambda d^2(j_1,j_2)}$ of the maximal distance $d(j_1, j_2) = \max_i \left\{||\mathbf{x}_{ij_1} - \mathbf{x}_{ij_2}||_2\right\}$ in the images in which both tracks are visible. The surface shape prior then is:

$$\mathscr{E}_{\mathrm{S}}(\mathsf{S}) = \sum_{(j_1,j_2)} \alpha(j_1, j_2) \cdot ||\mathbf{S}_{j_1} - \mathbf{S}_{j_2}||^2. \tag{7}$$

As for the smoothness prior we see that $\mathscr{E}_{\mathrm{S}}(\mathsf{S}) \neq \mathscr{E}_{\mathrm{S}}(\mathscr{A}^{-1}\mathsf{S})$.

6 Non-Rigid SfM With Priors

The model simultaneously minimizing the reprojection error, the smoothness prior and the surface shape prior, *i.e.* the cost:

$$\mathscr{E}_{\mathrm{RE}} + \gamma\mathscr{E}_{\mathrm{J}} + \beta\mathscr{E}_{\mathrm{S}} \tag{8}$$

must be obtained by nonlinear optimization. To ensure a good starting point, and because the coordinate frame in which the shapes are represented influences the solution, we choose (initially) this frame by minimizing the temporal smoothness prior. As shown below this fixes the mixing matrix up to an orthogonal matrix, to which the surface shape prior is invariant. Next, by using the surface shape prior an initial guess for S is estimated. Finally J and S are jointly refined by nonlinear least-squares optimization. The constants γ and β in (8) are chosen *ad hoc* such that the two priors initially contribute relative to the reprojection error with certain amounts, say 0.2 and 0.02. Below, the initial application of the two priors is described.

6.1 The Coordinate Frame

The prior measure (6) obviously depends on the mixing matrix. Consequently we (partially) determine this as the $r\times r$ full rank matrix $\mathscr{A}$ minimizing $\mathscr{E}_{\mathrm{J}}(\mathsf{J}\mathscr{A}) = ||\mathsf{L}\mathscr{A}||^2$. The motivation is that determining the mixing matrix ensures that the camera motion is 'close' to the optimal one. To avoid the shrinking effect of reducing the prior value by simply scaling down J we require $\det(\mathscr{A}) = 1$. Let $\mathsf{L} = \mathsf{U}\Sigma\mathsf{V}^{\mathsf{T}}$ be a (reduced) SVD of L. Below we sketch a proof for a closed-form solution for $\mathscr{A}$:

$$\mathscr{A} = \left(\sqrt[r]{\prod_{k=1}^{r}\sigma_k}\right)\mathsf{V}\Sigma^{-1}. \tag{9}$$

Given $\mathscr{A}$ we change the coordinate frame by $\mathsf{J} \leftarrow \mathsf{J}\mathscr{A}$ and $\mathsf{S} \leftarrow \mathscr{A}^{-1}\mathsf{S}$ without changing the reprojection error. However the value of the prior $\mathscr{E}_{\mathrm{J}}(\mathsf{J})$ is significantly reduced. It should be noted that (9) only fixes the mixing matrix up to a $r \times r$ orthogonal matrix.

A proof of equation (9). Let $\mathscr{A} = \mathsf{QDW}$ be an SVD of $\mathscr{A}$. We parameterize $\mathscr{A}$ as $\mathscr{A} = \mathsf{QD}$ since $\mathscr{E}_{\mathrm{J}}(\mathsf{J}\mathscr{A}) = \mathscr{E}_{\mathrm{J}}(\mathsf{JQD})$. Let $\mathsf{Y} = \mathsf{V}^{\mathsf{T}}\mathsf{Q} \in \mathscr{O}(r)$. We can rewrite $\mathscr{E}_{\mathrm{J}}(\mathsf{J}\mathscr{A})$ as:

$$||\mathsf{L}\mathscr{A}||^2 \;=\; ||\mathsf{U}\Sigma\mathsf{V}^{\mathsf{T}}\mathsf{QD}||^2 \;=\; ||\Sigma\mathsf{YD}||^2 \;=\; d_1^2||\Sigma\mathbf{y}_1||^2 + \cdots + d_r^2||\Sigma\mathbf{y}_r||^2 \qquad (10)$$

where $d_r \geq d_{r-1} \geq \cdots \geq d_1 \geq 0$ and with $\mathbf{y}_i$ the columns of Y. We want to find the $\mathbf{y}_i$ and the d_k minimizing the expression under the constraints that $\prod d_k = 1$, and that Y is orthonormal. Due to the ordering of the singular values we can split the minimization problem into r subproblems corresponding to the terms in the sum. From this we get $\mathsf{Y} = \mathrm{I}$, *i.e.* $\mathsf{Q} = \mathsf{V}$. The minimization problem then is reduced to:

$$\min_{\{d_k\}, \prod d_k = 1, d_r \geq \cdots \geq d_1 \geq 0} \sum_{k=1}^{r} (\sigma_k d_k)^2. \qquad (11)$$

Introducing Lagrange multipliers λ and μ_j a compound object function is formulated:

$$\min_{\{d_k\}} \sum_{k=1}^{r} (\sigma_k d_k)^2 + \lambda \left(\prod_{z=1}^{r} d_z - 1 \right) + \sum_{j=1}^{r} \mu_j (d_j - d_{j-1}). \qquad (12)$$

It can easily be shown that this function has a minimum given by:

$$2\sigma_k^2 d_k = \lambda \left(\prod_{z=1, z \neq k}^{r} d_z \right) = \frac{\lambda}{d_k}. \qquad (13)$$

Letting $\alpha = \sqrt{\lambda/2}$ and checking the unit determinant constraint it is seen that:

$$\alpha = \sqrt[r]{\prod_{k=1}^{r} \sigma_k}. \qquad (14)$$

Putting things together we reach expression (9).

To show that the minimum is global the Karush-Kuhn-Tucker conditions can be applied. A sufficient condition for the minimum to be global is that the three terms in (12) are twice differentiable and that the Hessian matrix evaluated in $\mathbb{R}^{r+}$ is positive semi-definite. The Hessian for the first term is diagonal with elements $2\sigma_k^2$. The last term is linear so the Hessian is a positive semi-definite null matrix. The Hessian for the second term $\prod_{z=1}^{r} d_z$ can easily be show to be positive semi-definite.

6.2 Surface Shape Prior Implementation

Having fixed the non-rotational part of the mixing matrix it becomes meaningful to compute an estimate of the structure S. Given the modified joint motion matrix J, S is sought to minimize a weighted sum of the reprojection error and the surface shape prior:

$$\mathscr{E}_{\mathrm{RE}} + \beta\mathscr{E}_{\mathrm{S}} \;=\; \|\mathscr{V} \odot (\mathsf{X} - \mathsf{JS} - \mathbf{t} \cdot \mathbf{1}^{\mathsf{T}})\|^2 + \beta \sum_{(j_1, j_2) \in \Omega} \alpha(j_1, j_2) \cdot ||\mathbf{S}_{j_1} - \mathbf{S}_{j_2}||^2 \qquad (15)$$

where $\mathscr{V}$ is the combined inlier and visibility matrix and Ω is the set of 'close' point tracks. The S minimizing this expression leads to a larger reprojection error compared to the initial solution. The reprojection error increases with β. We choose a value of β such that the increase in reprojection error is limited by a factor of 0.1 to 0.5. This is done using an iterative approach. Equation (15) can be rewritten as:

$$\mathscr{E}_{\mathrm{RE}} + \beta \mathscr{E}_{\mathrm{S}} \;=\; ||\mathbf{v} \cdot (\bar{\mathbf{x}} - \mathscr{M}\mathbf{s})||^2 + \beta ||\mathscr{L}\mathbf{s}||^2 \tag{16}$$

where $\bar{\mathbf{x}} = \mathrm{vect}(\bar{\mathsf{X}})$ and $\mathbf{s} = \mathrm{vect}(\mathsf{S})$. $\mathscr{M} = \mathrm{diag}_M(\mathsf{J})$ is a $(2NM) \times (rM)$ block diagonal matrix with M repetitions of J. If $p = |\Omega|$ is the number of 'close' pairs of tracks then $\mathscr{L}$ has p row blocks $\mathscr{L}_{(j_1,j_2)}$ of the form:

$$\mathscr{L}_{(j_1,j_2)} = \alpha(j_1, j_2) \cdot (0 \ldots 0, \mathrm{I}, 0, \ldots 0, -\mathrm{I}, 0 \ldots 0) \tag{17}$$

where I and 0 are the $r \times r$ identity and zero matrices, and where the positions of the two identity matrices correspond to the positions j_1 and j_2. Thus $\mathscr{L}$ will have the size $(rp) \times (rM)$. With this rewriting we can directly see that the least squares solution is

$$\mathbf{s} = [\mathscr{M}^\top \mathscr{M} + \beta \mathscr{L}^\top \mathscr{L}]^{-1} \mathscr{M}^\top \mathbf{x} \,. \tag{18}$$

7 Experimental Results

In the experiments reported below we concentrate on the improvement with respect to generalization by using the camera smoothness and surface shape priors.

7.1 Synthetic Data

In the first test we generated synthetic data with 100 frames and 100 point tracks and with true rank varying from 3 to 18. For each data set, models with and without use of the two priors were estimated from the diagonal 60% entries. The estimation error is measured as a function of the generalization distance in frames. For medium to large distances the error distribution were very long tailed. Therefore for each distance we measured the improvement in generalization by the ratio of medians without and with prior use. The generalization improvement measure increased with the rank as well as with the generalization distance. Figure 1 shows to the left the average (over all data sets) of the improvement. In more absolute terms we relate the error in the generalization area to the error in the training area by the percentage of points with generalization error exceeding a value $\mu + k\sigma$, where μ and σ are the mean and spread of the reconstruction error and $k = 2.5$, 5, and 10. An example is shown to the right in Figure 1. The smoothness measure (6) decreased by a factor between 80 and 500. The results on synthetic data showed that at the expense of a small increase of the reprojection error, the generalization error can be significantly reduced. In particular the number of very large errors is reduced. Experiments that are not reported here showed that the generalization improvement increased with the difficulty of the data, *e.g.* with the amount of measurement noise and with r.

7.2 Real Data

We applied the same testing procedure on data from two real sequences called *Bears* and *Groundhog day*. Figure 2 shows single frames from the two sequences. From the two

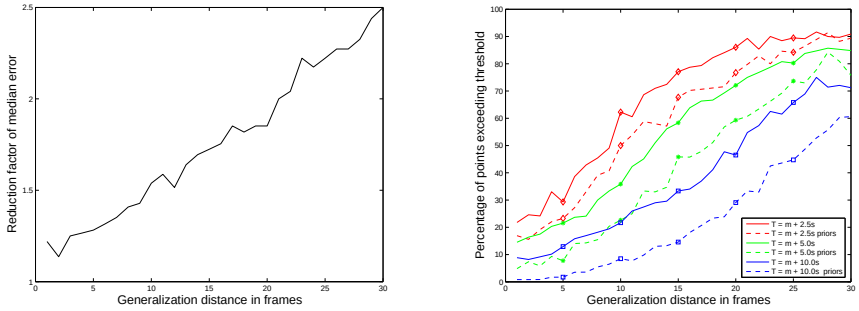


Figure 1: Results on synthetic data. Left: Average generalization improvement factor as a function of the generalization distance. Right: Percentage of point tracks with reprojection error exceeding three thresholds (see text), with and without prior use, as functions of the generalization distance.

Figure 2: Images from the 94-frames *Bears* sequence (left) and the 75-frames *Groundhog day* sequence (right) with marked points.

(originally banded) measurement matrices filled sub-matrices were extracted and a diagonal band with 50 % entries selected for training. The measurement matrices showed 94 and 75 frames with 94 and 117 point tracks. On the *Bears* sequence the camera smoothness measure was reduced by a factor of 108.7. The rank was estimated to 5. After initial estimation $\mathscr{E}_{\mathrm{RE}} = 0.82$ pixels. Applying the priors increased this to 1.20 pixels, a small payment for the improved generalization. Figure 3 shows plots of the percentage of point tracks as function of the generalization distance in frames, with and without use of the priors, and with reprojection error exceeding the previously described thresholds $\mu + k\sigma$, using $k = 2.5$, 5 and 10. Figure 3 shows that without prior use the generalization becomes bad even for short generalization distances. With prior use the error is significantly reduced. For the sequence *Bears* the generalization becomes possible at least up to a distance of 30 frames. On the more difficult sequence *Groundhog day* the camera smoothness measure was reduced by a factor of 5660.3. The rank was estimated to 14.

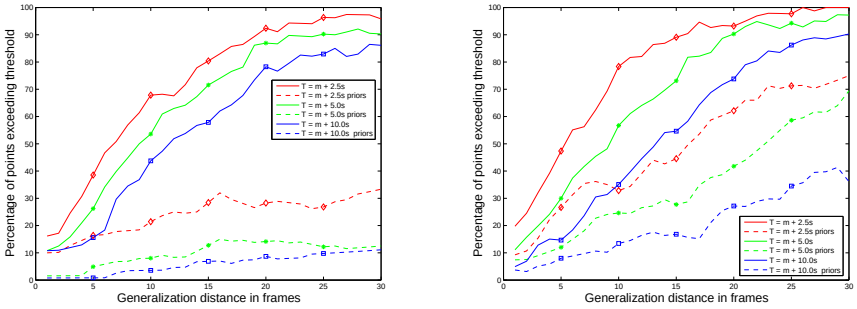


Figure 3: Percentages of point tracks in the sequences *Bears* (left) and *Groundhog day* (right) with reprojection error exceeding three thresholds (see text), with and without prior use, as functions of the generalization distance.

Figure 3 shows to the right that the generalization distance is increased by a factor of 2 to 4. This is still significant, but less impressive compared to the sequence *Bears*. A main reason is that a continuous surface is seen on the *Bears* sequence giving strength to the surface shape prior. This is not the case for the *Groundhog day* sequence.

In figure 4 a close-up of 4 tracks from the *Bears* sequence is shown. The positions

Figure 4: Close-up sequence of 4 point tracks which visible parts (use for training) all ended close to frame number 47. 'True' positions, given by the tracker, are shown by stars. Predicted positions estimated without using the priors are shown by diamonds. Predicted positions estimated with use of the priors are shown by squares.

computed by using the two are much closer to the true positions than the ones obtained by not using the priors.

8 Conclusions

We proposed an implicit non-rigid Structure-from-Motion approach with priors for temporal smoothness and surface shape coherency. We showed that the priors significantly improves the prediction of points in frames where data is missing, *i.e.* the generalization ability. Building on previous work the approach automatically estimates the rank of the measurement matrix, handles outliers and a substantial amount of missing data. Future

work will show if the improved generalization allows detecting and gluing point tracks split because of imperfect tracking. We expect the temporal smoothness prior to drive the estimated model closer to an explicit configuration. Further work how much this will help in such 'self-calibration'.

References

[1] H. Aanæs and F. Kahl. Estimation of deformable structure and motion. *The Vision and Modelling of Dynamic Scenes Workshop*, 2002.

[2] A. Bartoli and S. Olsen. A Batch Algorithm For Implicit Non-Rigid Shape and Motion Recovery. *Workshop on Dynamical Vision at ICCV'05*, 2005.

[3] M. Brand. Morphable 3D models from video. *Conf. on Computer Vision and Pattern Recognition*, 2001.

[4] M. Brand. A Direct Method for 3D Factorization of Nonrigid Motion Observed in 2D. *Conf. on Computer Vision and Pattern Recognition*, 2005.

[5] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3D shape from image streams. *Conf. on Computer Vision and Pattern Recognition*, 2000.

[6] A. M. Buchanan, A. W. Fitzgibbon Damped Newton Algorithms for Matrix Factorization with Missing Data. *Conf. on Computer Vision and Pattern Recognition*, 316–322, 2005.

[7] A. Del Bue, X. Lladó, L. de Agapito Non-Rigid Metric Shape and Motion Recovery from Uncalibrated Images Using Priors. *Conf. on Computer Vision and Pattern Recognition*, 1191–1198, 2006.

[8] D. W. Jacobs Linear Fitting with Missing Data for Structure-from-Motion. *Computer Vision and Image Understanding*, vol 82 no. 1, 57–81, 2001.

[9] D. Martinec and T. Pajdla 3D Reconstruction by Fitting Low-Rank Matrices with Missing Data. *Conf. on Computer Vision and Pattern Recognition*, pp. 198-205, 2005.

[10] C. Tomasi, T. Kanade: Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2), 137–154, 1992.

[11] P. H. S. Torr: Bayesian model estimation and selection for epipolar geometry and generic manifold fitting. *International Journal of Computer Vision*, 50(1):27–45, 2002.

[12] P. H. S. Torr, A. Zisserman: MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, vol. 78, 138–156, 2000.

[13] L. Torresani and C. Bregler. Space-time tracking. *European Conference on Computer Vision*, 2002.

[14] L. Torresani and A. Hertzmann. Automatic non-rigid 3D modeling from video. *European Conference on Computer Vision*, 2004.

[15] B. Triggs. Linear projective reconstruction from matching tensors. *Image and Vision Computing*, 15(8), 1997.

[16] R. Vidal and D. Abretske. Nonrigid Shape and Motion from Multiple Perspective Views. *European Conference on Computer Vision*, 2006.

[17] J. Xiao, J.-X. Chai, and T. Kanade. A closed-form solution to non-rigid shape and motion recovery. *European Conference on Computer Vision*, 2004.

[18] J. Xiao and T. Kanade. Non-rigid shape and motion recovery: Degenerate deformations. *International Conference on Computer Vision and Pattern Recognition*, 2004.

Automated Analysis of Deformable Structure in Groups of Images

V Petrović, T Cootes, C Twining, A Mills and C Taylor
Imaging Science and Biomedical Engineering
University of Manchester
M13 9PT, UK
v.petrovic@manchester.ac.uk
http://www.isbe.man.ac.uk

Abstract

We describe an approach for automated analysis of deformable objects which extracts structure information from groups of images containing different examples of the object with a particular application to human imaging. The proposed analysis framework simultaneously segments and registers a set of images, incrementally constructing a model of the composition of the object. By fitting an appropriate intensity distribution model to the image we obtain a soft segmentation which allows us to explicitly model the construction of each pixel from constituent image segments, rather than its expected intensity. This effectively decouples the model from the effects of the imaging system and varying statistics in different examples. When estimating the optimal deformation field for each example, the original image is compared to a reconstruction, generated using the composition model and its intensity distribution parameters for each segment (i.e. an estimate of how the model would appear given the imaging conditions for that image). In the paper we describe the algorithm in detail and show results of applying it to two sets of medical images of different anatomies taken with different imaging modalities. We present quantitative results demonstrating that the proposed algorithm is more powerful than current state of the art methods at extracting structural information such as spatial correspondences across groups of images with varying statistics.

1 Introduction

This paper proposes an automated approach for analysing, understanding and representing deformable object structure in groups of images, with a particular application to medical imaging and biometrics. The human body is an abundant source of objects that share a common anatomical structure but exhibit an almost infinite number of shape and appearance variations. Generally, given a set of images of different examples of an object with a deformable structure, we would like to derive in an automated manner (without user intervention) the following:

- a dense spatial and structural correspondence between the various examples (registration)

- a consistent composition of the pixels in each example image into different constituent parts of the structure (segmentation)
- a statistical representation of the variability of shape and appearance of the structure across the set (modelling)

Furthermore, an explicit advantage would be if all of the above could be achieved in an efficient and robust (converging) manner. There has already been considerable research into techniques that aim to reach each of the listed goals independently. Non-rigid image registration, and in particular *groupwise* methods provide a method of deriving a dense, spatial correspondence across sets of images [10, 1] (for a review see [14]). Direct segmentation of medical images, into different tissues for example, has also been studied extensively with methods based on pixel intensity and more advanced deformable structures [8, 13]. Finally, Statistical Shape and Appearance Models [4], are capable of capturing and describing the appearance (shape and texture). variation of the modeled structure.

A number of other works exploit the fact that a good estimate of any one aspect of the structure, a correct segmentation, registration or a good model, can help derive more reliable estimates of the other components. For instance combined segmentation and registration with active contours was considered in [12] to register single objects. Maximum a posteriori segmentation using hidden Markov random fields and B-spline non-rigid registration was used for more general medical images [2]. Models of deformation have been constructed from correspondences estimated by non-rigid registration [7, 9], but it was also shown that it is possible to integrate modelling and registration more tightly [5].

In this paper we describe an automated approach which combines simultaneous segmentation, registration and modeling of structure in a single iterative framework to satisfy the requirements laid out above. The method starts with a training set of images and incrementally constructs a model of the composition of each pixel in the common structure, rather than its expected intensity. This decouples the model from details of the imaging process and modality and allows us to deal with datasets exhibiting significant variation in intensity. Extensive qualitative and quantitative results demonstrate that the proposed algorithm is more powerful than current state of the art methods at extracting structural information such as spatial correspondences across groups of images with varying statistics.

The method is described in detail in Section 2 while results of applying it to two sets of medical images of different anatomies taken with different imaging modalities, digital radiography (X-ray, DR) and magnetic resonance (MR), are provided in Section 3. Finally we provide a discussion on the relative merits of the presented approach.

2 Method

An overview of the proposed approach is illustrated in Figure 1 showing example images from an application of the approach to MR images of the human brain. Generally, a set of N images T_i, $i = 1...N$, (the training set) is assumed to contain a common structure that consists of M distinct components whose content is defined according to some composition model F and whose intensities obey some specific distribution model with parameters θ_i. Furthermore, for the entire set, a spatial correspondence with a reference

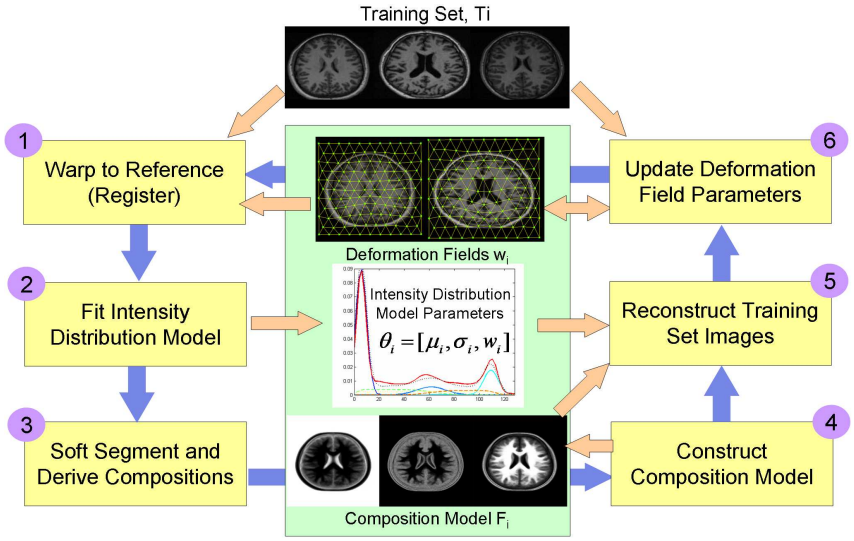


Figure 1: Outline of the proposed structure analysis algorithm: dark arrows indicate the progress of the algorithm, light arrows flow of data and the central box contains the structural information derived from the data set

(model) frame, and implicitly with each other, is assumed through a set of spatial deformation fields defined for each example in the training set, $W_i()$. Deformations are initialised as identity transformations and true correspondences $W_i()$ along with the intensity distribution model parameters θ_i and the structure composition model F are then estimated incrementally across the set in an iterative procedure as follows:

1. Warp each training image T_i into the reference frame using the current estimate of the deformation field. $T_i' = W_i^{-1}(T_i)$.
2. Fit the intensity distribution model to each image and extract parameters (means, SDs and weights) for each of the M components encoded in $\theta_i = \{\mu_{ij}, \sigma_{ij}, w_{ik}\}$), as well as distributions due to mixtures of components.
3. Use the resulting distributions to estimate the most probable composition of each pixel, and encode a set of fraction images $F_i^{(j)}$, $j = 1...M$ for each training example.
4. Combine the fraction images from all examples to construct a single composition model for the common structure, $\{\hat{F}^{(1)}...\hat{F}^{(M)}\}$.
5. Synthesize a reconstruction of each training set image S_i using the current estimates of intensity distribution parameters θ_i (μ_{ij}) and the current composition model $\hat{F}$
6. Update the current estimate of W_i to best register S_i onto T_i, minimising a suitable similarity measure, $D_{im}(T_i, W_i(S_i))$.

The stages listed above are repeated in an iterative procedure until the deformation field optimisation and the composition model converge. The reference frame defining the model shape is obtained as the mean of all individual shapes, represented through W_i. Initial identity deformation fields will contain a considerable misalignment of the examples

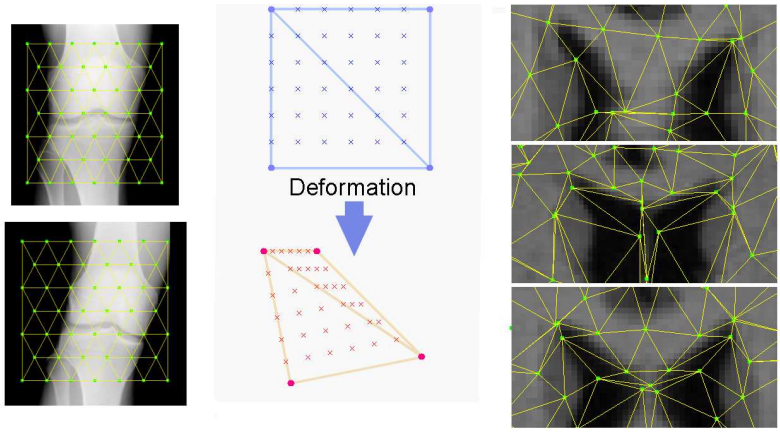


Figure 2: Piecewise affine deformation field: identity fields on two X-ray images (left), linear interpolation of a deformed shape (middle) and converged fields on corresponding areas in three MR brain images

resulting in a fuzzy composition model. However, as the algorithm progresses and correct correspondences become established both the composition model and the model shape will converge to a true, crisp representation of the underlying structure.

Note that the described process involves no construction of a shape model. Instead an explicit statistical appearance model of the structure can be constructed directly at the end of the process from the converged $W_i()$, see [5].

2.1 Establishing Correspondence

Spatial correspondence between the examples is established by defining a deformation field for each image in the training set that defines where each pixel on the reference structure is located on that image. This implicitly imposes a structural correspondence that allows equivalent locations to be found across the examples. We adopt a piece-wise affine deformation field represented as a tesselation (triangulation in 2D) of a set of control points (vertices) in space, Figure 2. Deformation is controlled by displacement of the control points, which can be both linear (e.g. affine) and highly non-linear (movements of individual points). Inside the elements the field is interpolated linearly, Figure 2, which lends efficiency and more importantly easy invertibility to this formulation at a price of limited spatial resolution and flexibility.

Deformation fields are initialised in 2D as a regular hexagonal mesh made up of equilateral triangles, see Figure 2, which provides a regular element density around each control point as opposed to a square regular mesh. The fields are then optimised in discrete stages that modify the locations of control points either in groups or individually. The details of the optimisation strategy are beyond the scope of this paper, but the general approach is to start with linear transformations (e.g. affine), followed by coarse non-rigid deformations, e.g. grid deformations [5] and progressively increase the resolution of the deformations to finish by optimising the location of each control point independently.

2.2 Segmentation

A broad segmentation of the analysed structure is achieved in two stages. First an intensity distribution model (IDM) is fit to the intensity histogram of the data in the reference frame and then a most likely composition of each pixel in each example is derived using IDM parameters. The IDM explains how the intensities in the image are related to the main components of the structure to be analysed. In principle any type of distribution model can be used within this framework but it is likely that each type of data would optimally obey a specific model. As the choice of the intensity model for a particular dataset is not central to the structure of the proposed algorithm it is not considered in detail in this paper. Instead we use relatively simple models that rely only on intensity and demonstrate the convergence power of the approach.

In general we follow [8] in assuming that each pixel in the structure is either due to one of M different components or a fractional mixture of at most two different ones. Furthermore, if we know the distributions of intensities for pure components, we can construct the distribution for a particular fractional distribution by convolution. For example, in the experiments using MR images we use a limited resolution IDM that assumes components with Normal distributions, $p_i(g) = N(g : \mu_i, \sigma_i^2)$ (consistent with white matter, grey matter and cerebro-spinal fluid/background tissue types). In this case it can be shown that the distribution for a partial volume with fraction f of tissue type i and $1-f$ of type j is given by

$$p_{ij}(g|f) = N(g : f\mu_i + (1-f)\mu_j, f\sigma_i^2 + (1-f)\sigma_j^2). \tag{1}$$

The distribution over all partial volumes containing i and j is given by

$$p_{ij}(g) = \int_{f=0}^{f=1} p_{ij}(g|f)p(f)df = \int_{f=0}^{f=1} p_{ij}(g|f)df \tag{2}$$

where we assume all values of f in the range $[0,1]$ are equally likely ($p(f) = 1$). Making the assumption that any pixel contains at most 2 different tissue types, we need only consider M pure tissue classes with distributions $p_k(g)$, $k = 1..M$, and $M(M-1)/2$ partial tissue classes (enumerated $p_k(g)$,$k = (M+1)..M_t = M(M+1)/2$). Thus the measured image intensity distribution, $h(g)$, can be approximated as a weighted sum

$$p(g : \theta) = \sum_{i=1}^{M_t} w_i p_i(g) \tag{3}$$

where $\theta = \{\mu_i, \sigma_i, w_k\}$ ($i = 1..M, k = 1..M_t$).

We thus perform an optimisation to estimate the parameters θ which optimise $D_p(p(g : \theta), h(g))$, where $D_p(p,q)$ is a suitable measure of divergence between distributions. Having estimated the probability that a pixel with intensity g belongs to class k is given by $P_k(g) = w_k p_k(g)/(\sum w_k p_k(g))$ (see Figure 3) that pixel can then be classified as belonging to class

$$k_c = \arg\max_k P_k(g). \tag{4}$$

However, we are actually interested in the estimate of the fraction of each pure class tissue (f_i,$i = 1..M$), in the pixel, not the probability of each class. If $k_c \leq M$ then the pixel is a pure tissue, so we define $f_{k_c} = 1$ and $f_{i \neq k_c} = 0$. If $k_c > M$ then the pixel is classified

as a partial volume, containing two tissues, say of type i and type j. In this case we wish to find the most likely value of the fractions for each tissue. We define

$$\begin{aligned} f_i &= \arg\max_f p_{ij}(f|g) \\ &= \arg\max_f p_{ij}(g|f)p(f)/p(g) \\ &= \arg\max_f p_{ij}(g|f) \end{aligned} \tag{5}$$

where $p_{ij}(g|f)$ is defined above in Equation 1. We then set $f_j = 1 - f_i$ and $f_{k\neq i,j} = 0$. Figure 3 shows an example of this, demonstrating that tissue probabilities are not the same as estimates of pure tissue fractions. Using this approach we compute M images, $\{F_i^{(1)}, ...F_i^{(M)}\}$, recording the fraction of each tissue type at each pixel in the normalised version of image i (that projected into the reference frame).

2.3 Composition Model Construction

The composition model defines how much of each of the components is present at any location within the structure that is being analysed. We train this model using the M fractional images from each of the N images in our set. [1] Though more detailed statistical models (eg PCA based methods) are possible, in this preliminary study we simply compute the mean of the fraction images,

$$\{\hat{F}^{(1)}...\hat{F}^{(M)}\} = \frac{1}{N}\sum_i \{F_i^{(1)}...F_i^{(M)}\}. \tag{6}$$

Further constraints could be imposed on the model, e.g. limit any pixel to have at most two non-zero fractions. Although this would directly support convergence, particularly in the early stages of the process when misalignments between different examples are still considerable we found that even the simple mean was proving powerful enough to drive the process to convergence.

2.4 Image Reconstruction

The training set is aligned by optimising a deformation field between each T_i and the model (reference frame) embodied in a reconstruction, S_i produced using the current composition model and the current estimate of the IDM parameters. Pure components exhibiting Gaussian distributions are optimally represented by their mean (μ_{ij}) while fractional pixels are represented by a sum of component means weighted by their fractions:

$$S_i = \sum_{j=1}^{M} \mu_{ij}\hat{F}^{(j)}. \tag{7}$$

For an example, see Figure 3. Essentially, S_i is an estimate of how the model would appear given the imaging conditions for T_i. Ideally S_i is a noise free version of T_i but in practice it starts blurred due to misalignments and gets progressively sharper as alignment across the set improves. Deformation parameters W_i are optimised with respect to an objective function measuring similarity between T_i and S_i in the training image frame - $D_{im}(T_i, W_i^{-1}(S_i))$.

[1] In practice, when working on image i constructing the model from $N-1$ other images tends to give more generalisable models and lead to faster convergence.

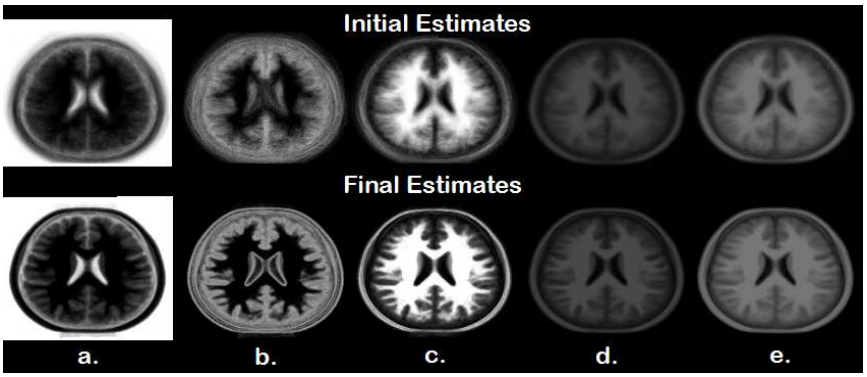


Figure 3: Results of analysis of brain images: a.-c. composition model estimates for the three tissue classes (components), d. and e. reconstructed images of two training set examples (reference frame)

3 Results

We applied the proposed method to two sets of medical images of different anatomies taken with different imaging modalities, a set of 28 X-ray digital radiography (DR) images of the knee joint and of a set of 37 near equivalent 2D slices of magnetic resonance (MR) images of the brain (Figure 3) [2]. For the X-ray images we adopted an absorption IDM which has 2 classes (no radiation and full radiation) at extremes of the intensity range represented as delta Diracs and all intensities in between are considered fractional. In addition we used sum of absolute differences for both the image similarity, $D_{im}()$, and Bhattacharya distance as the divergence between intensity distributions, $D_p()$.

Figure 3 a.-c. shows composition models for the three (tissue) components present in the MR brain images processed by the proposed method. In the final estimates all three classes are crisply delineated and in close agreement with the anatomical distribution of white and gray matter (WM, GM) and CSF in the human brain. Structure reconstruction images S_i corresponding two different training set examples are shown in Figure 3 d. and e. It can be seen that their intensity statistics have been reproduced faithfully by the algorithm. In both cases, the composition model starts from a very fuzzy estimate and becomes more accurate as the alignment across the training set examples is established.

Figure 4 shows the results of analysis on the knee X-rays. This is a difficult data set containing projections of a structure with highly unconstrained pose, scale and image statistics, see 4e. Groupwise intensity registration [5] fails to converge resulting in a mean image 4b, very much like the mean of the non-aligned set 4a. The proposed approach however converges and its mean 4c. clearly shows the main structures. Final absorption (composition) model is shown alongside in 4d. Final deformation fields for three different examples produced by the proposed algorithm are shown in 4e. They demonstrate its ability to deal with large variations in pose and intensities robustly and converge despite the fact that some examples have diverged during affine registration (final example). These failures are caused by the generally sparse structure of these images failing to con-

[2] David Kennedy of the Center for Morphometric Analysis, Boston, provided the MR and Visaris d.o.o. provided the DR imagery

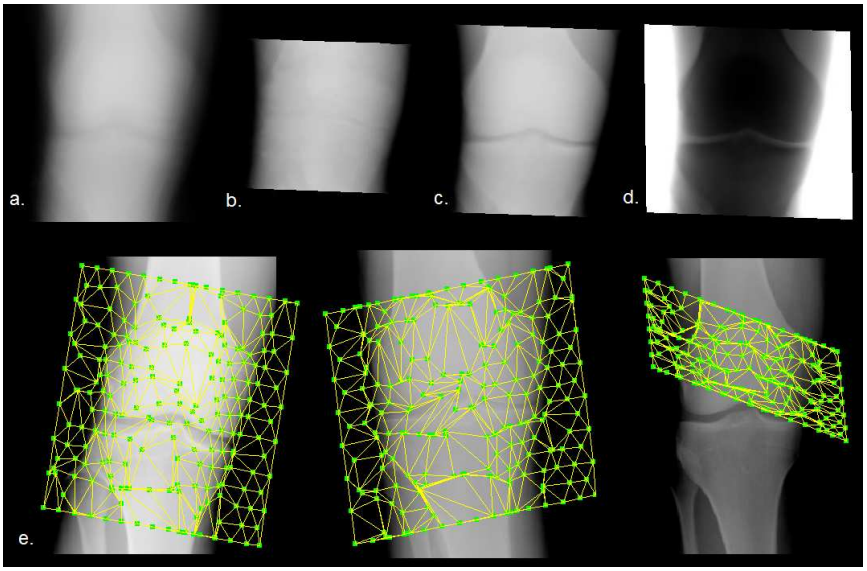


Figure 4: Results of automated analysis of knee images: a. initial mean, b. mean derived using groupwise intensity registration, c. mean derived using the proposed approach and associated composition model for full radiation d., e. final deformation fields for three different images using the proposed approach

strain a powerful global search such as affine registration and could be corrected using relatively straightforward regularisation across the set.

Quantitative evaluation was performed on WM, GM and ventricle (CSF) labels manually defined by experts on the MR brain data using a Tanimoto overlap based metric proposed in [6] (no such ground truth was available for the knee images). The metric measures fuzzy overlap of segmented regions between all pairs of registered images in the set. Results for inverse volume normalised ($TO_{IVol-All}$) [6] and mean of pairwise overlaps for individual as well as all labels (TO_{Label}) are in Table 1. The proposed automated analysis framework (AAF) system was compared to i) pairwise registration where each image in the set is registered to a common reference image selected either randomly PW-random or one closest to the mean of the set PW-opt, ii) groupwise registration where the set is registered to its progressively sharper intensity mean [5] (all using 24x24 point piece-wise affine deformation field and sum of absolute differences objective function) and iii) fluid flow registration (Fluid) [3], using a dense deformation field (defined at each pixel), sum-squared difference objective function, viscosity coefficients $\lambda = 1$ and $\mu = 500$, tolerance for convergence 1e-3, two levels of scale and time step selected by Brent minimization.

Table 1 shows that the proposed algorithm outperforms other systems for all metrics and labels. Figure 5a. shows these results graphically (TO_{all}) including measurement errorbars as well as final intensity means for the PW-opt, GW and the proposed approaches in comparison to the initial mean. Also shown on Figure 5a. as the dashed line is the $TO_{All} = 0.717$ level obtained for groupwise registration of label images, in a way establishing an upper limit on the performance for the chosen registration approach (de-

Metric	PW-rand	PW-opt	GW	Fluid	AAF
$TO_{IVol-All}$	0.591	0.61	0.646	0.651	**0.69**
TO_{All}	0.603	0.616	0.652	0.635	**0.693**
TO_{WM}	0.662	0.664	0.696	0.684	**0.747**
TO_{GM}	0.551	0.537	0.59	0.578	**0.633**
$TO_{Ventricle}$	0.596	0.664	0.669	0.685	**0.69**

Table 1: Quantitative label overlap scores for registration results of various approaches applied to the MR brain data (best score given in bold)

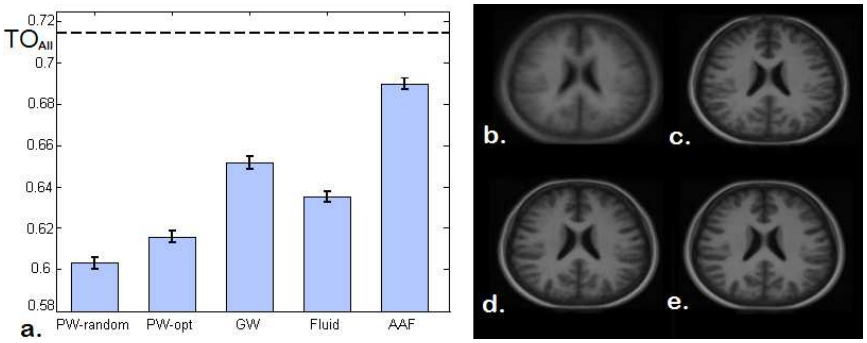


Figure 5: MR brain analysis resuts: a. Label overlap (TO_{All}) results for various approaches, b. initial (non-aligned) mean intensity of the images and c-e. final intensities for the PW-opt, GW and proposed AAF approaches

formation field representation and optimisation scheme). The proposed method achieves overlaps only 2% lower than this limit and much closer than any of the other methods using the same registration approach (in comparison the equivalent reference value for the tested fluid registration approach is $TO_{All} = 0.672$).

4 Discussion

We have demonstrated a powerful algorithm for automated analysis of deformable structure in groups of images. By constructing a model of structure composition, rather than intensities, we decouple the model from details of the imaging process, and concentrate on explicitly learning object structure. The system should be capable of registering images from different modalities. In evaluations on two challenging datasets the proposed framework outperforms other state-of-the-art approaches, despite relying on relatively simple intensity models for segmentation and a relatively coarse deformation field representation.

Future work will include a full implementation to deal with full 3D structures (the extension is natural) and exploring robust segmentation that includes spatial as well as local gradient information. Further consideration will also be given to automating he optimal choice of intensity models for a given dataset, using approaches such as MDL [10, 11] as well as derivation of generic models capable of dealing with various types of objects and image data.

References

[1] K. K. Bhatia, J. V. Hajnal, B. K. Puri, A. D. Edwards, and D. Rueckert. Consistent group-wise non-rigid registration for atlas construction. *Proceedings of the IEEE Symposium on Biomedical Imaging (ISBI)*, pages 908–911, 2004.

[2] D. Rueckert C. Xiaohua, M. Brady. Simultaneous segmentation and registration for medical image. In *Proceedings of MICCAI 2004*, number 3216 in Lecture Notes in Computer Science, pages 663 – 670, 2004.

[3] G Christensen, R Rabbitt, and M Miller. Deformable templates using large deformation kinematics. *IEEE Trans. Medical Imaging*, 5:1435 – 1447, 1996.

[4] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models – their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[5] T.F. Cootes, C.J. Twining, V. Petrovic, R. Schestowitz, and C.J. Taylor. Groupwise construction of appearance models using piece-wise affine deformations. In *Proceedings of the* 16^{th} *British Machine Vision Conference (BMVC)*, volume 2, pages 879–888, 2005.

[6] W Crum, O Camara, and D Hill. Generalised overlap measures for evaluation and validation in medical image analysis. *IEEE Trans. Medical Imaging*, 25:1451 – 1461, 2006.

[7] A. F. Frangi, D. Rueckert, J. A. Schnabel, and W. J. Niessen. Automatic construction of multiple-object three-dimensional statistical shape models: Application to cardiac modelling. *IEEE Transactions on Medical Imaging*, 21(9):1151–1166, 2002.

[8] N. A. Thacker M. Pokric and A. Jackson. The importance of partial voluming in multi-dimensional medical image segmentation. In *Proceedings of Information Processing in Medical Imaging (IPMI)*, volume 2208 of *Lecture Notes in Computer Science*, pages 1293–1294. Springer, 2001.

[9] D. Rueckert, A. F. Frangi, and J. A. Schnabel. Automatic construction of 3D statistical deformation models using non-rigid registration. *Lecture Notes in Computer Science*, 2208:77–84, 2001.

[10] C. J. Twining, T. F. Cootes, S. Marsland, V. S. Petrovic, R. S. Schestowitz, and C. Taylor. A unified information-theoretic approach to groupwise non-rigid registration and model building. In *Proceedings of Information Processing in Medical Imaging (IPMI)*, volume 3565 of *Lecture Notes in Computer Science*, pages 1–14. Springer, 2005.

[11] C. Twining V. Petrović, T. Cootes and C. Taylor. Automatic framework for medical image registration, segmentation and modeling. In *Proceedings of Medical Image Understandind and Analysis (MIUA)*, pages 141–145, 2006.

[12] A. Yezzi, L. Zollei, and T. Kapur. A variational framework for inregrating segmentation and registration through active contours. *Medical Image Analysis*, 7:171–185, 2003.

[13] Y. Zhang, M. Brady, and S. Smith. Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm. *IEEE Transactions on Medical Imaging*, 20:45 – 57, 2001.

[14] Barbara Zitová and Jan Flusser. Image registration methods: A survey. *Image and Vision Computing*, 21:977 – 1000, 2003.

Distribution-based Level Set Segmentation for Brain MR Images

Jundong Liu[1], David Chelberg[1], Charles Smith[2], Hima Chebrolu[2]
[1]School of Elec. Engi. & Comp. Science
Ohio University, Athens, Ohio
[2]Department of Neurology
University of Kentucky, Lexington, KY

Abstract

In this paper, we propose a distribution-based active contour model for brain MRI segmentation. As a generalization of the Chan-Vese piecewise-constant model, our solution uses Bayesian a posterior probabilities as the driving forces for curve evolution. Distribution prior, if available, can be seamlessly integrated into the level set evolution procedure. Unlike other region-based active contour models, our solution relaxes the global piecewise-constant assumption, and uses locally varying Gaussians to better account for intensity inhomogeneity and local variations existing in many MR images. More accurate and robust segmentations are therefore achieved. Experiments conducted on synthetic and real brain MRIs demonstrate the improvement made by our model.

1 Introduction

Magnetic resonance imaging (MRI) is a rich source of information regarding the soft tissue anatomy of human brains. Segmentation of Magnetic resonance imaging (MRI) brain images into different tissue types, i.e., gray matter (GM), white matter (WM) and cerebrospinal fluid (CSF) is a critical and fundamental task for the large volume of 3D MRI data to be effectively utilized for disease diagnosis, functional analysis of brains and the treatment of disease related to brain anamolies.

A variety of approaches to brain MRI segmentation have been proposed in the literature. Histogram-based approaches estimate the probability of a class label given only the intensity for each voxel. Such estimation problems are usually formulated in the sense of maximum a posteriori (MAP) or maximum likelihood (ML) estimates. With respect to the form of the probability density function, finite Gaussian mixture models [12] are assumed and used in segmentation.

Recently, segmentation algorithms [15, 2, 3, 10, 13, 5] that use region-based active contour models have gained great popularity. Active contour without edge model, commonly known as Chan-Vese piecewise-constant model [2], uses a stopping term based on Mumford-Shah segmentation functional so that the model can detect object boundaries with or without gradient. Although impressive experimental results have been reported for this model and its variants [10, 5] some common drawbacks and limitations exist within this group of solutions. A *mixture of global Gaussians* (piecewise-constant can

be regarded as the degenerate case) has been used a convenient assumption for modeling the intensity distribution. *Global means* are utilized to discriminate regions from each other. However, "homogenous regions with distinct means" is rarely an accurate account in practice, especially for medical images. In addition, spatial distribution priors, often available and being used extensively in histogram-based models, are normally neglected in the region-based active contour models. Prior knowledge about the organ's location sometimes is an indispensable resource to separate certain tissue types from their surroundings.

Non-parametric region-based active contours models (Chan-Vese [3] and Tsai-Yezzi [13]) can theoretically handle the local intensity variation problem. In their algorithms, an image u_0 is modeled by piecewise smooth functions u^+, u^- that are defined inside and outside a closed active contour, respectively. The curve evolution is carried out through an iterative process. In each iteration, u^+ and u^- are estimated first, by solving a Poisson equation with Neumann boundary condition. Then, the level set function is updated following a gradient flow that minimizes the simplified Mumford-Shah functional. Unlike in the parametric models, images in the piecewise smooth framework (both Chan-Vese and Tsai-Yezzi) are modeled as a smooth random field within each region. Intensity variability thus can be handled across regions without the need to specify the change on statistical parameters.

However, with the burden to solve a Poisson PDE in each iteration, piecewise-smooth models suffer from inevitable high computational costs induced from solving certain huge sparse linear system. Being computationally expensive has been a major obstacle for these models to be used in practical 3D medical applications [8].

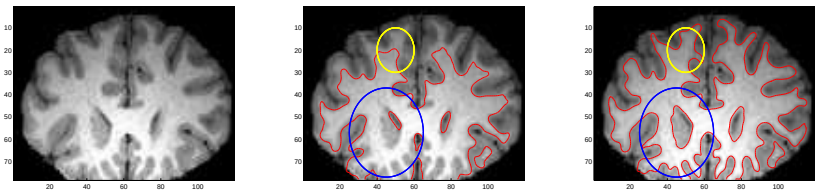
1.1 Our proposed solution

Aiming to reap the benefits and avoid the drawbacks of the piecewise-constant and piecewise-smooth models, we propose a bridging solution in this paper. To generalize the Chan-Vese model, we adopt Bayesian a posterior probabilities as the driving force for the curve evolution. Our model has two desired properties: 1) distribution prior can be seamlessly integrated into the level set evolution procedure and leads to more robust segmentations; 2) piecewise constant assumption is relaxed from "global" to "local", and *local means* are used as the area representatives. Being able to better account for intensity inhomogeneity, our model works particularly well for the images with low intensity contrasts and spatially varying brightness variations. When the computation switches from global to local, segmentation "twisting" (same objects are labeled oppositely at different local areas) may happen if no global control is in place. We tackle this issue with a selective update scheme, which enforces a global-to-local consistency over the entire image domain.

2 Methods

Let C be an evolving curve in Ω. C_{in} denotes the region enclosed by C and C_{out} denotes the region outside of C. Chan-Vese (two-phase) piecewise-constant model is to minimize the energy functional

$$F(c_1,c_2,C) \quad = \quad \mu \cdot \text{Length}(C) + \lambda_1 \int_{C_{in}} |u_0 - c_1|^2 dxdy + \lambda_2 \int_{C_{out}} |u_0 - c_2|^2 dxdy$$



(a) (b) (c)

Figure 1: Chan-Vese model's inability to handle local image variations. a) is a slice of brain MRI image before bias correction. b) is the curve evolution result using Chan-Vese model. c) is the result using our method.

where c_1 and c_2 are the averages of u_0 inside C and outside C respectively.

This model has several attractive properties: 1) it is very robust to weak boundaries and noise; 2) interior contours can be automatically recovered; 3) the initial curve can be anywhere in the image; 4) it has few parameters for adjustment compared to MRF-based methods; 5) Efficiency wise, re-initialization of the level set function often is not required, and big step size can usually be taken for level set update.

These appealing advantages, however, are not easily utilizable to the full extent in practice. Global Gaussian distribution assumption are not an accurate depiction of local image profile for many medical images. Negligence of local information would often result in undesired segmentations. Figure 1 shows an example where the Chan-Vese piecewise-constant model fails to produce an expected segmentation result. Fig 1.a) is an MR image with bias field. The bias field lights up gradually from the top to the bottom of the image. Due to this intensity variation, the global means c_1 and c_2 can not represent the image well, and undesired segmentation, as highlighted in Fig 1.b), is resulted. (The figure is better seen on screen than in print)

Piecewise-smooth models [3, 13] provide a solution for the intensity variability problem. Gradual intensity changes, as in the Fig. 1 can be handled with [3, 13], however, high computational cost and being sensitive to curve initialization pose a barrier for practical applications.

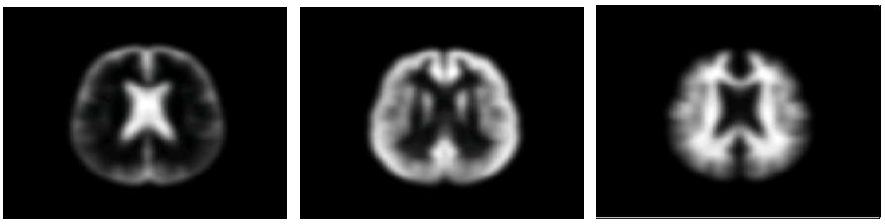
2.1 Our Local Distribution-based Model

Let $S = \{in, out\}$ be the two classes for a two-phase model. The probability of the pixel (x,y) belonging to *in* and *out* is denoted by $P(in|(x,y))$ and $P(out|(x,y))$ respectively. Let $Pr(in)$ and $Pr(out)$ be the class prior probabilities at (x,y). Then,

$$P(in|(x,y)) = \frac{Pr(in_{(x,y)})P(u_0(x,y)|in)}{P(B)} \quad P(out|(x,y)) = \frac{Pr(out_{(x,y)})P(u_0(x,y)|out)}{P(B)} \quad (1)$$

where $P(u_0(x,y)|in)$ is the likelihood of a voxel in class *in* has the intensity of $u_0(x,y)$. $P(B)$ is a constant. Bayesian decision rule states that $u_0(x,y)$ should be classified into the class *in* if:

$$Pr(in_{(x,y)})P(u_0(x,y)|in) > Pr(out_{(x,y)})P(u_0(x,y)|out)$$



(a) (b) (c)

Figure 2: Spatial prior probability images of CSF, GM and WM.

or otherwise into *out*. If a perfect segmentation/classicition is achieved, this inequality should hold for each voxel (x,y), if every pixel has been classified into the correct class. Based on this observation, we can formulate the segmentation problem as the minimization of the following energy:

$$\begin{aligned} F(C) &= \mu \cdot \text{Length}(C) - \int_{C_{in}} \log(Pr(in)P(u_0(x,y)|in))dxdy \\ &\quad - \int_{C_{out}} \log(Pr(out)P(u_0(x,y)|in))dxdy \end{aligned}$$

Note that our overall model is similar to [10, 11], but the setup of the likelihood term is different, which will be explained next.

2.1.1 Spatial distribution priors: $Pr(in)$ and $Pr(out)$

Many distribution prior images have been generated from recent brain studies [4]. A widely used model is provided by the Montreal Neurological Institute [7] as part of the ICBM, NIH P-20 project. MNI prior is made of three probability images that contain values in the range of zero to one, representing the prior probability of a voxel being either GM, WM or CSF after an image has been normalized to the same space (see Figure 2). In this paper, we are particularly interested in extracting sub-cortical GM, therefore we take the GM and WM prior images as $Pr(in)$ and $Pr(out)$ respectively, for demonstration purpose. For these prior images to be applied, a registration is need to align the prior and the input image. We used the affine registration routine provided by SPM [12] in all the 3D experiments of this paper.

2.1.2 Likelihood terms: global Gaussian versus local

As illustrate in Fig. 1, global Gaussians and global means are not an accurate description of the local image profile, especially when intensity inhomogeneity is present. A remedy is to relax the global Gaussian mixture assumption and take local intensity variations into consideration. More specifically, local Gaussians (local binary as the degenerate case) should be used as a better approximation to model the vicinity of each voxel.

In the Chan-Vese model, two global means c_1 and c_2 are computed for C_{in} and C_{out}. In our approach, we introduce two functions $v_1(x,y)$ and $v_2(x,y)$, both defined on the image domain, to represent the mean values of the *local* pixels inside and outside the moving curve. By *Local*, we mean that only neighboring pixels will be considered. A simple

implementation of the "neighborhood" is to introduce a rectangular window $W(x,y)$ with size of $2k+1$ by $2k+1$, where k is a constant integer. Therefore,

$$v_1(x,y) = \text{mean}(u_0 \in (C_{in} \cap W(x,y)))$$
$$v_2(x,y) = \text{mean}(u_0 \in (C_{out} \cap W(x,y)))$$

With the new setup, our segmentation model can then be updated as a minimization of the following energy:

$$F(v_1,v_2,C) = \mu \cdot \text{Length}(C) - \int_{C_{in}} \left(\log(Pr(in)) - \log(\sigma_1) - \frac{(u_0-v_1)^2}{2\sigma_1^2} \right) dxdy - \int_{C_{out}} \left(\log(Pr(out)) - \log(\sigma_2) - \frac{(u_0-v_2)^2}{2\sigma_2^2} \right) dxdy$$

The variances σ_1 and σ_2 should also be defined and estimated locally. However, due to the fact that local variance estimation tends to be very unstable, we use global variances (for the pixels in C_{in} and C_{out}) as uniform approximation.

2.2 Level set framework and gradient flow

Using the Heaviside function H, and the one-dimensional Dirac measure δ [2] as the bridge, the energy function $F(v_1,v_2,C)$ can be minimized under the level set framework. Let $T_1 = \log(Pr(in))$ and $T_2 = \log(Pr(out))$, and we have the following new functional to minimize:

$$F(v_1,v_2,C) = \mu \int_\Omega \delta(\phi|\nabla\phi|)dxdy - \int_\Omega \left(T_1 - \log(\sigma_1) - \frac{(u_0-v_1)^2}{2\sigma_1^2} \right) H(\phi)dxdy - \int_\Omega \left(T_2 - \log(\sigma_2) - \frac{(u_0-v_2)^2}{2\sigma_2^2} \right) (1-H(\phi))dxdy$$

Under the level set framework, we deduce the associated Euler-Lagrange equation for the level set function ϕ. Parameterizing the descent direction by an artificial time $t \geq 0$, the gradient flow for $\phi(t,x,y)$ is given as

$$\frac{\partial\phi}{\partial t} = \delta(\phi) \left[\mu \text{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) - \log\frac{Pr(in)}{Pr(out)} + \log\frac{\sigma_1}{\sigma_2} - \left(\frac{(u_0-v_1)^2}{2\sigma_1^2} - \frac{(u_0-v_2)^2}{2\sigma_2^2} \right) \right] \quad (2)$$
$$\phi(0,x,y) = \phi_0(x,y) \text{ in } \Omega$$

where ϕ_0 is the level set function of the initial contour. This gradient flow is the evolution equation of the level set function of our proposed method.

Correspondingly, v_1 and v_2 are computed with

$$v_1 = \frac{(u_0 * H(\phi)) \otimes W}{H(\phi) \otimes W} \qquad v_2 = \frac{(u_0 * (1-H(\phi))) \otimes W}{(1-H(\phi)) \otimes W} \quad (3)$$

where $\otimes$ is the convolution operator. One should note that, Chan-Vese model can be regarded as a special case of our model — when the window W is set to infinitely large.

In practice, the Heaviside function H and Dirac function δ in eqn. 3 have to be approximated by smoothed versions. We adopt the $H_{2,\varepsilon}$ and $\delta_{2,\varepsilon}$ used in [2]. For all the experiments conducted in this paper, we set the size of the window W as 21×21.

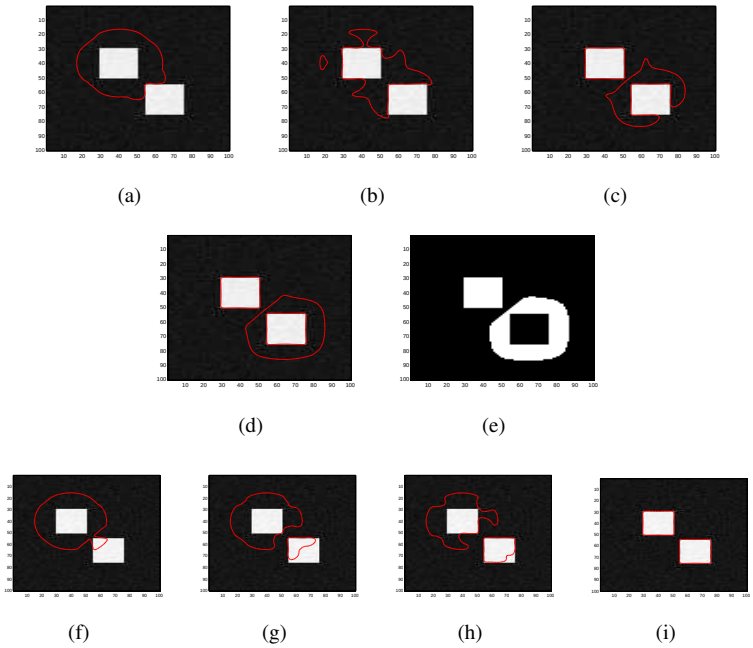


Figure 3: Illustration of the occurrence of "local twists". a), b), c) and d) are four snapshots of the level set propagation. e) is the resulted segmentation. The effectiveness of the control term is illustrated in (f-i), which are four snapshot of the level set propagation of the new gradient flow. (The figures are better seen on screen than in black-white print)

3 Global-to-Local Consistency Constraint

In Chan-Vese piecewise-constant model, as the entire image is considered as a whole, the signs of the level set function ϕ correspond very well to the segmented classes. In other words, if certain class S has more than one components, at the time a perfect segmentation is achieve, each of them would be enclosed at the same side of ϕ. The positive side (ϕ^+) and the negative (ϕ^-) side of ϕ, partition the image domain into two homogeneous regions.

However, under our proposed local Gaussian environment, this property is not guaranteed. Since the level set function ϕ evolves based on $v_1(x,y)$ and $v_2(x,y)$ that are computed locally, the multiple components of a same class might be evolved into the opposite sides of ϕ, therefore labeled with different classes. We give a name to this phenomena as *local twisting*. An example in Fig. 3 illustrates how a *twist* occurs. The evolving curve starts as a circle covering part of the left square. As v_1 and v_2 are computed locally, it happens that the left half the level set function ϕ goes up, and the right half goes down. Eventually the left square is enclosed under ϕ^+ and the right square under ϕ^-. The two squares are expected to classified into the same class, but the evolution based on Eqn.3 sends them into two different groups, as shown in Fig 3.b. The phenomenon is due to the lack of global control over the evolution process. Whenever *local twisting* happens, incorrect segmentation will be resulted.

Assume we use ϕ^+ to capture the brighter portion of a bimodal image. In order to eliminate *local twists*, the following consistency constraint needs to be enforced everywhere in the image domain:

Constraint: $v_1(x,y) \geq v_2(x,y)$**, for all** $(x,y) \in \Omega$

There would be many different implementations to enforce this constraint, and we find the following approach particularly effective and simple:

Solution: use sign$(v_1(x,y) - v_2(x,y))$ **as a control term to guide the update of** ϕ.

where $\text{sign}(x) = 1$, if $x > 0$ and $\text{sign}(x) = 0$ otherwise.

At the locations where no twist is present, $v_1 > v_2$, this control term $\text{sign}(v_1(x,y) - v_2(x,y))$ would let the level set update as Eqn.3 specifies. At certain locations, if $v_1(x,y) < v_2(x,y)$ happens, the control term put a halt to the level set update at (x,y), and further development of a potential twist is avoided. Through this mechanism, twists are controlled at an early stage, and will eventually disappear when the normal configuration $(v_1 > v_2)$ dominates over the image domain.

The above analysis, together with the solution, also applies to the case that we use ϕ^+ to capture the darker object. Putting the above analysis together, the updated gradient flow for our model is modified to:

$$\frac{\partial \phi}{\partial t} = \text{sign}(v_1 - v2) \cdot \delta(\phi) \left[\mu \text{div}(\frac{\nabla \phi}{|\nabla \phi|}) - \log \frac{Pr(in)}{Pr(out)} + \log \frac{\sigma_1}{\sigma_2} - \left(\frac{(u_0 - v_1)^2}{2\sigma_1^2} - \frac{(u_0 - v_2)^2}{2\sigma_2^2} \right) \right] \quad (4)$$

$$\phi(0,x,y) = \phi_0(x,y) \text{ in } \Omega$$

4 Results and Discussions

The fist experiment we conducted is based on the image shown in Fig 1.a. We tried to segment this 2D brain image into GM and WM. Since no prior information is available, we set $\log(Pr(in))$ and $\log(Pr(out))$ both to 0.5. Our result is shown in Fig 1.c, along with that of Chan-Vese model in Fig1.b. It is evident that our method can capture the local details, and produces a very accurate segmentation.

The second example is another MR image with bias field. Due the existing bias field, this image greatly violates the *global Gaussian/mean* assumption, therefore traditional region-based approaches, including the Chan-Vese model, are expected to fail. Figure 4 shows the result of using Chan-Vese model (left column) and that of using our local median model (right column). Three snapshots of the executions are provided. As evident, Chan-Vese model has trouble in capturing the GM area in the top-left and right-bottom corners, while our model separate the two issues very accurately.

The third experiment is based on the same MR image, but with an added artificial bias field. The result is shown in Fig 5. The purpose of the added field is to test how well our new model in handling severe intensity variation. Owing to the tremendous amount of inhomogeneity, piecewise-constant model totally failed, while our model still works very well without being affected by the bias level. This experiment also serves as a very good indication of the robustness of our approach.

The last group experiments were conducted on seven 3D MR images. All subjects are participants in the longitudinal University of Kentucky Alzheimer's Disease Center

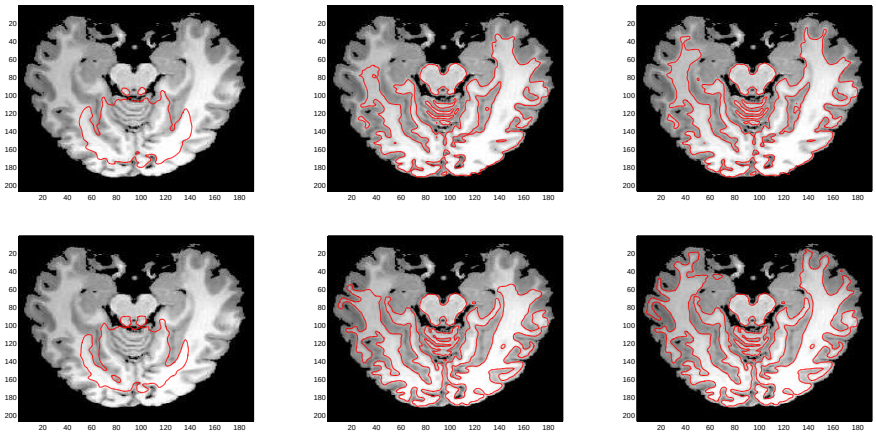


Figure 4: Segmentation comparison of Chan-Vese model and our model in handling bias field. First row: three snapshot of the execution on Chan-Vese model; Second row: three snapshot for our model. (The figures are better seen on screen than in print)

"biologically resilient adults in neurological studies" (BRAiNS) group. Scanning was performed on a Siemens Vision 1.5T instrument. We compared our solution with that of SPM [12] and Chan-Vese model. Fig. 6 shows a single slice result from all three methods. Fig.6.a is the input image, and 6.b, 6.c and 6.d are the GM segmentation from SPM, Chan-Vese and our model, respectively. The sub-cortical GM tissues in all the seven images have a bit higher intensity values than cortical GM, therefore the Chan-Vese model, using a piece-wise constant assumption, mis-classifies quite a portion of putamen as WM. Our model, on the other hand, clearly separates the putamen and thalamus from their surrounding WM. The comparison for the sub-cortical area has been highlighted with a red circle in Fig.6 (Figures are better seen on screen than in black-white print). Spatial distribution prior and local Gaussians both play a role in achieving this improvement. Compared to SPM, our model has the edge in outlining cleaner cortical GM (highlighted with a blue circle; better seen on the screen). Since level set methods all generate binary segmentations, our model can be used as a discrete alternative for SPM.

References

[1] K. V. Leemput et al., Automated model-based tissue classification of MR images of the brain", IEEE Trans. on Medical Imaging, vol. 18, pp. 897-908, 1999

[2] T. F. Chan and L. A. Vese, Active contours without edges, IEEE Trans. on Image Processing, Vol. 10, No. 2, pp. 266-277, 2001.

[3] T. F. Chan, L. A. Vese, A level set algorithm for minimizing the Mumford-Shah functional in image processing, 1st IEEE Workshop on Variational and Level Set Methods in Computer Vision, pages 161-168, 2001.

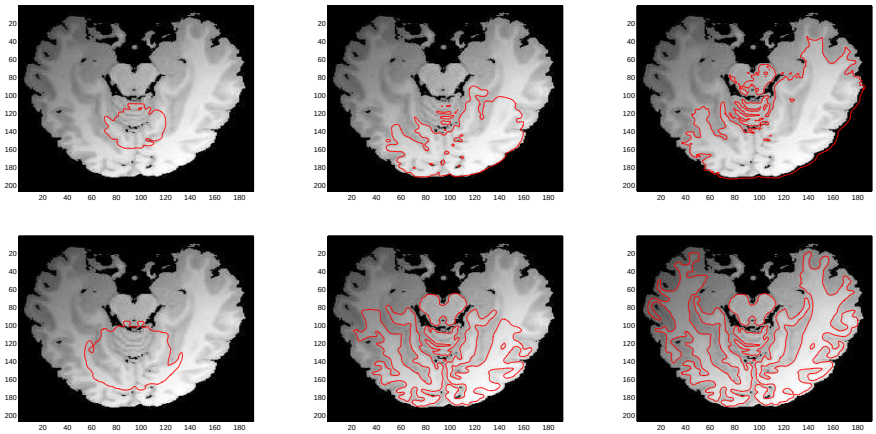


Figure 5: Segmentation comparison of Chan-Vese model and our model in handling severe intensity inhomogeneity. First row: three snapshot of the execution on Chan-Vese model; Second row: three snapshot for our model. (The figures are better seen on screen than in print)

[4] C. A. Cocosco et al., BrainWeb: Online interface to a 3D MRI simulated brain database, Neuroimage, vol. 5, no. 4, part 2/4 S245, 1997

[5] D. Cremers, M. Rousson and R. Deriche, "A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape", IJCV, 2006. To appear.

[6] J. Yang, H. Tagare, L. H. Staib, J. S. Duncan, "Segmentation of 3D Deformable Objects with Level Set Based Prior Models". ISBI 2004: 85-88.

[7] A. C. Evans, D. L. Collins and B. Milner, "An MRI-based stereotactic atlas from 250 young normal subjects", Society of Neuroscience Abstrasts, 18:408, 1992.

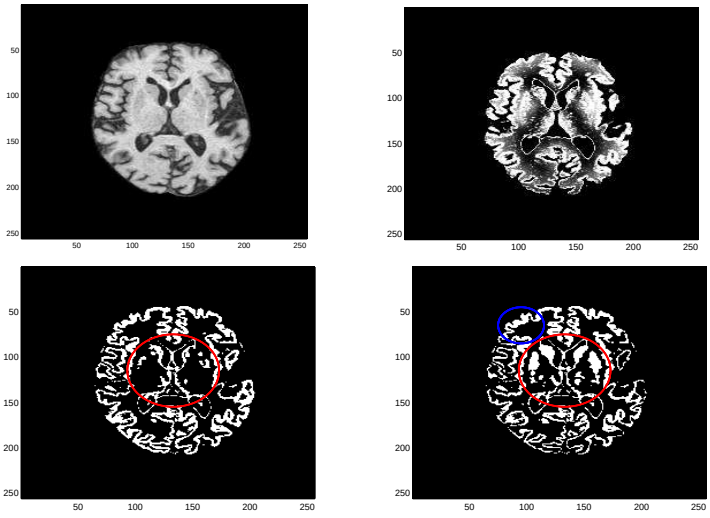
[8] S. Gao, T. D. Bui, Image Segmentation and Selective Smoothing by Using Mumford-Shah Model. IEEE Transactions on Image Processing 14(10), 1537-1549, 2005.

[9] C. Li, J. Liu, M. D. Fox: Segmentation of Edge Preserving Gradient Vector Flow: An Approach Toward Automatically Initializing and Splitting of Snakes. CVPR (1) 2005: 162-167.

[10] N. Paragios and R. Deriche, "Coupled Geodesic Active Regions for Image Segmentation: A Level Set Approach", ECCV (2) 2000, pp. 224-240.

[11] M. Rousson, R. Deriche, "A Variational Framework for Active and Adaptative Segmentation of Vector Valued Images", INRIA Technical Report, 2002.

[12] A. Mechelli, C.J. Price, K.J. Friston, and J. Ashburner. "Voxel-Based Morphometry of the Human Brain: Methods and Applications". Current Medical Imaging Reviews, pp 105-113, 2005.



(c) (d)

Figure 6: Input image and 3 GM segmentation results from SPM (b), Chan-Vese (c) and our model (d).

[13] A. Tsai, A. Yezzi, W. Wells, C. Tempany, D. "Approach to Curve: Evolution for Segmentation of Medical Imagery", IEEE TMI, Vol. 22, No. 2, 137-154, February 2003

[14] C. Xu and J. L. Prince, "Snakes, Shapes, and Gradient Vector Flow," IEEE Transactions on Image Processing, 7(3), pp. 359-369, March 1998.

[15] S. Zhu and A. Yuille, "Region competition: Unifying snakes, region growing, and bayes/MDL for multiband image segmentation", PAMI, 18(9):884–900, 1996.

Sparse MRF Appearance Models for Fast Anatomical Structure Localisation*

René Donner[12], Branislav Mičušík[2],
Georg Langs[31], Horst Bischof[1]
[1]Institute for Computer Graphics and Vision, Graz University of Technology, Austria, bischof@icg.tugraz.at
[2]Pattern Recognition and Image Processing Group, Vienna University of Technology, Austria
{donner,micusik}@prip.tuwien.ac.at
[3]GALEN Group, Laboratoire de Mathématiques Appliquées aux Systèmes, Ecole Centrale de Paris, France, georg.langs@ecp.fr

Abstract

Image segmentation methods like active shape models, active appearance models or snakes require an initialisation that guarantees a considerable overlap with the object to be segmented. In this paper we present an approach that localises anatomical structures in a global manner by means of *Markov Random Fields (MRF)*. It does not need initialisation, but finds the most plausible match of the query structure in the image. It provides for precise, reliable and fast detection of the structure and can serve as initialisation for more detailed segmentation steps.

*Sparse MRF Appearance Models (*SAMs*)* encode a priori information about the geometric configurations of interest points, local features at these points and local features along the edges of adjacent points. This information is used to formulate a Markov Random Field and the mapping of the modeled object (e.g. a sequence of vertebrae) to the query image interest points is performed by the MAX-SUM algorithm.

The local image information is captured by novel symmetry-based interest points and local descriptors derived from *Gradient Vector Flow*. Experimental results are reported for two data-sets showing the applicability to complex medical data.

1 Introduction

The reliable and fast detection and segmentation of anatomical structures is a crucial issue in medical image analysis. It has been tackled by a number of powerful approaches,

*We would like to thank Philipp Peloschek, MD and Klaus Friedrich, MD of the Department of Radiology, Medical University of Vienna, Austria, for supplying the medical images. This research has been supported by the Austrian Science Fund (FWF) under grant P17083-N04 (AAMIR), as well as the European Union Network of Excellence FP6-507752 (MUSCLE) and the Region Île-de-France.

among them active shape models [3], active appearance models [4, 5], active feature models [12], graph-cuts [2] and snakes [7].

These approaches have been successfully employed to segment structures in cardiac MRIs [16] or for registration in functional heart imaging [19]. In [17] vertebrae in the spine were delineated, and in [20] active shape models were utilised for bone densiometry.

All approaches rely on a reasonable initialisation of the iterative active appearance model or active shape model search: ASMs and AAMs need to be placed with considerable overlap with the object of interest. Graph-cuts need a set of manually annotated seed points placed within and outside of the object, and while snakes need spatial constraints, to ensure the delineation of the correct object. Usually the initialization is either done manually or by application specific approaches.

Several approaches to a detect coarse initialization positions rely on pair-wise point matching using local descriptors like SIFT [13], shape context [1] or PCA-SIFT [8], and typically rely on a robust method like RANSAC [6] to deal with ambiguous structures. They match interest points between a source (i.e. example) image and the until now unseen target image. These approaches have several drawbacks: (1) For complex non-rigid transformations between source and target image a large number of correct interest points matches is required, which increases computation time considerably for the robust matching. (2) Information about the spacial relation of adjacent descriptors is difficult to incorporate into the matching process.

In this paper we propose a deterministic method based on Markov Random Fields (MRF) that incorporates both interest point positions and local features to perform the detection of landmark configurations from a single example. The detection is performed in a fast manner by the MAX-SUM algorithm [21]. The approach uses all interest point features and positions and finds a solution which minimizes the combined costs of non-rigid deformations and local descriptor feature differences. Arbitrary interest points and local descriptors can be used. We report results for interest points based on local symmetry and a complementary local descriptor derived from gradient vector flow [22].

Local symmetry detectors were investigated in [15, 10], but they are either computationally expensive or use radial symmetry detection with predefined radii. Recently [14] proposed an approach to detect symmetry in the constellation of interest points detected by existing point detection methods.

The paper is structured as follows: In Sec. 2 we explain the interest point detector and local descriptor. Sec.3 details Markov Random Fields and in Sec. 4 the mapping of the source- to the target points by MRFs are explained in detail. In Sec. 5 we present the experimental evaluation of our approach, followed by a conclusion and an outlook in Sec. 6.

2 Symmetry based interest points and descriptors

Many structures of interest to medical experts, like bones, veins and many anatomical structures or their parts exhibit a shape with a high degree of symmetry w.r.t. one or more axes. This property of (local) symmetry is well preserved even when dealing with 2D slices of 3D data sets like MRIs, as the cross sections of these body parts will appear as round or elongated structures. Even regions of interest that do not exhibit this property can

1082

be localized by observing their neighborhood, e.g. an initialization for e.g. meniscoids can be provided by correctly localizing the discs and vertebrae of the spine.

2.1 Interest Points from Local Symmetry

Popular interest point detectors which are often used in conjunction with SIFT are the Harris corner detector and the Difference of Gaussians (DoG) approach, neither of which possess an affinity to local symmetry. A comparison of the interest points detected by DoG and interest points derived from local symmetry is shown in Fig. 1 (a,b).

To detect points of high local symmetry we use the gradient vector flow (GVF) field, which was originally proposed in [22] to increase the capture range of active contours. Its strengths include the ability to detect even weak structures while being robust to high amounts of noise in the image. The GVF can be computed either from a binary edge map or directly from the gray level image $\mathbf{I}$. We compute the GVF of an image as $\mathbf{G} = u + i * v = GVF(\mathbf{I})$, yielding the complex matrix $\mathbf{G}$ used for all subsequent computations. The resulting field $\mathbf{G}$ is depicted in Fig. 2 for a synthetic example and a section of a hand radiograph, overlaid over the image $\mathbf{I}$.

The field magnitude $|\mathbf{G}|$ is largest in areas of high image gradient, and the start- and endpoints of the field lines of $\mathbf{G}$ are located at symmetry maxima. E.g. in the case of a symmetrical structure formed by a homogeneous region surrounded by a different gray level value the field will point away form or towards the local symmetry center of the structure, as shown in Fig. 2 (a,b). The symmetry interest points are thus defined as the local minima of $|\mathbf{G}|$. In contrast to techniques based on estimating the radial symmetry using a sliding window approach this will yield a sparse distribution of interest points even in large homogeneous regions.

After detecting the interest points the orientation $\alpha_i \in [0, 2\pi[$ of the local region surrounding the interest point can be estimated. Around each interest point rays $\mathbf{g}^r_\alpha$ at the 360 angles $\alpha \in [0, \ldots, 2\pi[$ at radii $r \in \{2, \ldots, 8\}$ are sampled from $|\mathbf{G}|$ using bilinear interpolation. The interest point i is then assigned the angle α_i which minimises

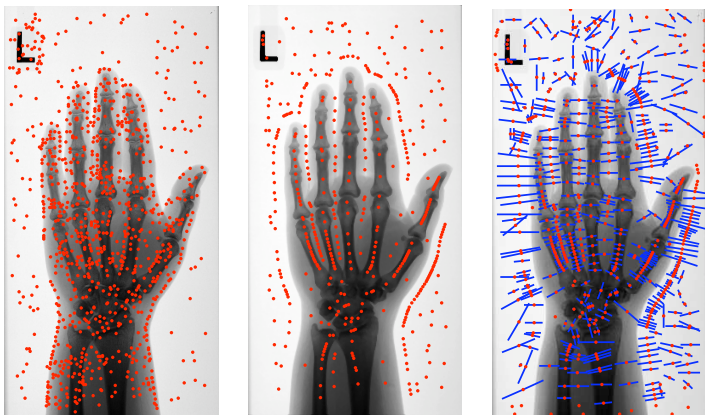
$$\alpha_i = \underset{\alpha \in [0, 2\pi[}{\operatorname{argmin}} \sum_r \mathbf{g}^r_\alpha. \tag{1}$$

The scale s_i of the region around the interest point is estimated by the mean distance of the interest point i to the two closest local maxima of $|\mathbf{G}|$ in the directions of α_i and $\alpha_i + \pi$. Examples for the resulting estimates for orientation and scale are shown in Fig 1 (c). If the scale varies only within a limited range as for the medical images examined in this paper the scale can remain fixed.

2.2 Local Descriptors from Gradient Vector Flow Fields

A measure is needed to specify the similarity of the local regions around the symmetry interest points and edges. Several local descriptors have been proposed in recent years, including SIFT [13] and Shape Context [1]. While most of these approaches yield descriptors suitable for building the MRF, they would require additional computations. In contrast, we can directly use $\mathbf{G}$ to describe local context.

In [8] normalized patches of the image gradient are used, extracted according to the interest points' orientation and scale as local descriptor are. Similarly, we extract patches



(a) (b) (c)

Figure 1: Comparison of the (a) interest points found by Difference of Gaussians (DoG) and (b) the symmetry points found as minima of GVF magnitude. Note how the symmetry points pick up the structures which are of interest to medical experts, greatly facilitating the correct localization of these structures. (c) depicts the scale and orientation estimates obtained around the symmetry points.

of $\mathbf{G}$ around the symmetry interest points, according to scale s_i and orientation α_i, as depicted in Fig. 2. They are re-sampled to a 10×10 grid and the vector field's orientations are stored relative to α_i to form the actual local descriptor. This encodes the information about the image gradients within and around the patch in a rotation-invariant way. Because of the GVF's smooth structure, Euclidean distance can be used used to compute the distance between two descriptors. This eliminates the need for complex histogram construction as performed by SIFT for example, while still retaining a feature vector of low dimensionality.

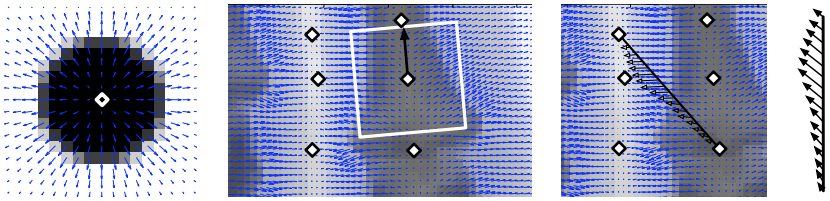
As the orientation of the local interest point is usually only stable up to $\pm\pi$, the actual distance between two local descriptors $\mathbf{D}_1$ and $\mathbf{D}_2$ is computed as $min(\|\text{abs}(\mathbf{D}_1 - \mathbf{D}_2)\|, \|\text{abs}(\mathbf{D}_1 - \mathbf{D}_2^*)\|)$, where $\mathbf{D}_2^*$ denotes the descriptor 2 rotated by π.

Local edge descriptors In addition to the local descriptors around interest points the appearance along the models' edges forms an important part of sparse appearance models.

Again the GVF $\mathbf{G}$ is used to extract the relevant information. Given 2 interest points in the image, $\mathbf{G}$ is sampled at equidistant points along the edge. The sampled field is then stored relative to the edge's orientation, forming a complex vector $\mathbf{e}$ as displayed in Fig. 2 (d). For the experiments in this paper, 40 points were sampled per edge.

By describing an image using the GVF-based local descriptors around interest points and along edges, the essential information about the structure of the anatomical object is captures in a sparse fashion. Sec. 4 describes how the descriptors from several training images are combined to form a sparse appearance model.

1084



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

Figure 2: (a) Examples of GVF with the detected symmetry interest points (diamonds). (b) Descriptor extraction from the GVF field. Around each symmetry point patches are extracted from the vector field according to their scale and orientation. The patch is then resampled to a 10×10 grid, relative to the interest points' orientation, to form the actual descriptor. The image is displayed for better visualization, the symmetry points are marked as diamonds. (c) Schematic edge descriptor of the edge between two points, formed by sampling the GVF at equidistant points along the edge. (d) Resulting edge descriptor, relative to the edge's orientation.

3 Markov Random Fields and the MAX-SUM problem

The Markov Random Fields considered in this paper represent graphs where each of the M nodes, called objects, has N fields, or labels, with associated qualities. The labels of two adjacent nodes are fully connected by N^2 edges, again with a weight to encode quality. Which objects are adjacent is encoded in an additional graph $\mathscr{A}$ with A edges. This basic structure is depicted in Fig. 3 (a). There are 4 objects with 3 labels each, with $N^2 = 9$ edges between the adjacent objects, A is 5.

Of interest is now to select one label for each object, so that the sum of label and edge qualities of the resulting sub-graph becomes maximal, illustrated as thick lines. The MAX-SUM solver can be used to tackle this problem. The MAX-SUM (labeling) problem of the second order is defined as maximizing a sum of bivariate functions of discrete variables. The solution of a MAX-SUM problem corresponds to finding a configuration of a Gibbs distribution with maximal probability. It is equivalent to finding a maximum posterior (MAP) configuration of an MRF with discrete variables [21].

Let the $M \times N$-matrix $\mathbf{C}$ represent the label qualities for each of the objects, and the $A \times N^2$-matrix $\mathbf{E}$ represent the edge qualities between the pairs of labels.

The total quality of the label selection $\mathbf{S} = \{n_1, \ldots, n_M\}$ with $n_i \in \{1, \ldots, N\}$ is then defined as

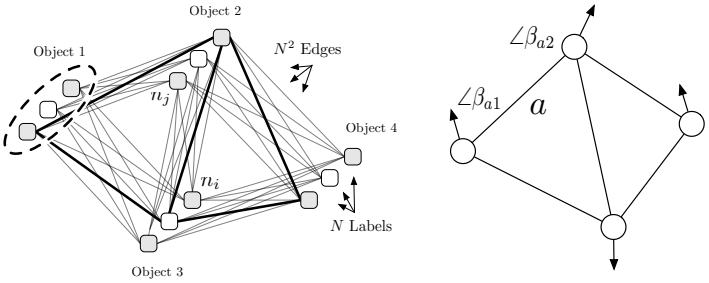
$$C(\mathbf{S}) = \sum_{m=1\ldots M} \mathbf{C}(m, \mathbf{S}(m)) + \sum_{a=1\ldots A} \mathbf{E}(a, \beta(E, \mathbf{S}, a)), \tag{2}$$

where $\beta(\mathbf{E}, \mathbf{S}, a)$ denotes the column representing the quality of the edge between the labels chosen to represent the edge $\mathscr{A}(a)$.

Solving the MAX-SUM problem means finding the set of optimal labels

$$\mathbf{S}^* = \underset{\mathbf{S}}{\operatorname{argmax}}\, C(\mathbf{S}). \tag{3}$$

Recently, a very efficient algorithm for solving this problem through linear programming relaxation and its Lagrangian dual, originally proposed by Schlesinger in 1976 [18],



(a) (b)

Figure 3: (a) The MRF graph consists of M objects with N labels each. Qualities are assigned to both labels and edges. Finding the solution to a MAX-SUM problem means selecting a label for each object, such that the sum of qualities of the selected labels and the edges connecting them is maximized. (b) Illustration of how the relative angles between an edge and the orientations of its adjacent vertices is computed.

has been presented [21].

The MAX-SUM solver permits several labels to be defined while still keeping the processing time within reasonable bounds. There are other attempts to solve the labeling problem for MRF using, e.g., second order cone programming [11], sequential tree-reweighted max-product message passing [9] or belief propagation methods [23]. However, neither of the algorithms, nor the MAX-SUM approach, solve the problem of a multi-label MRF exactly, as it is NP-hard. If the graph is a tree the global optimum of Eq. (3) is guaranteed [9], in the case of a non-tree graph MAX-SUM takes various approximations into account to reach a possibly optimal solution.

4 Sparse Appearance Model Matching

This sections describes how the sparse appearance model is constructed from training data. This model is then used to specify the Markov Random Field for a target image.

Building a Sparse Appearance Model Sparse appearance models extract information from images using local descriptors around interest points and along the edges between these points. No PCA based model is used to avoid the need for a large number of training samples and the global character of PCA-based models. The shape model is based on a Delaunay triangulation of the model points, and statistical models of the edges' lengths, relative angles and local descriptors are recorded. This yields a locally deformable rotation invariant model. The interest points and local point/edge descriptors are based on local symmetry and GVF as described in Sec. 2.

For each of the n model images, a subset of M interest points is manually selected to describe the anatomical structure to be found. One of the model images is used to define the graph structure using a Delaunay triangulation of its M model points. The resulting adjacencies of model points yield the set $\mathscr{A}$ of index-tuples describing the edges. Examples of the generated model are shown in Fig. 4 (a,b).

1086

The M selected model points represent the objects of the MRF graph, while the N target interest points correspond to the labels. A solution $\mathbf{S}$ thus represents a mapping of the model interest points to a subset of the target interest points.

We now need to build the a priori statistical models from the n training samples for the M model points and the A edges between these model points. First the orientations of the model points are normalized. As the n training orientations for a model point m are only stable up to $\pm\pi$, π is added to a subset of them such that the circular variance of the n orientations of model point m is minimised.

As there are generally too few training samples to estimate the parameters of a multivariate Gaussian in the space of the local descriptors, only the mean of the n local descriptors for each model point m is used, yielding descriptors $\overline{\mathbf{D}}_m$.

For each edge a of the A model edges, the mean length $\bar{l}_a$ and the standard deviation l_a^σ is computed. Similarly, from the n angles β_{a1} and β_{a2} between the edge and the orientations of its vertices the mean angles and standard deviations $\overline{\beta}_{a1}$,$\overline{\beta}_{a2}$ and β_{a1}^σ,β_{a2}^σ, computed using circular statistics, are stored (see Fig. 3 (b)). The third edge property which is modelled is the local descriptor (see Fig. 2). Similarly to the point descriptors, the mean descriptor $\bar{\mathbf{e}}_a$ is computed for each model edge.

Constructing the MRF Given a sparse appearance model and a target image, the Markov Random field is used to model the confidences that a model point or edge should be matched to a certain interest point or edge in the target image. As we are solving a maximization problem, all confidences or qualities are in the interval $[-\infty, 0]$. The descriptor distances are normalized to having a maximum of 0 and a median of -1, while the length and angle confidences are $\in [-1, 0]$.

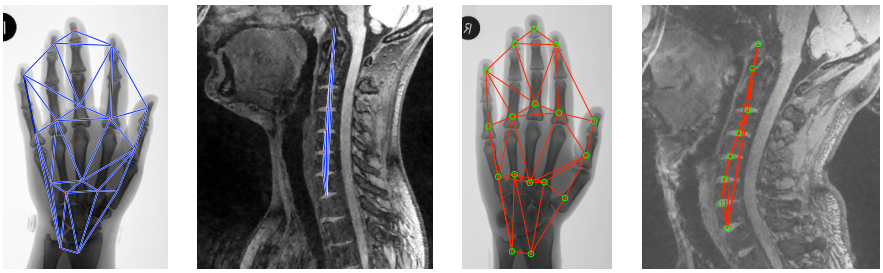
The quality of a (model point, target point)-match equals the negative distance between the local target descriptor and the model point descriptor $\overline{\mathbf{D}}_m$. All mutual distances between model and potential target correspondences are computed, resulting in the $M \times N$-matrix $\mathbf{C}$.

The qualities of the AN^2 edges in the model are stored in $\mathbf{E}$. The quality of an edge e between two labels n_i, n_j in $\mathbf{E}$ is computed by comparing its length l_e and relative angles β_{e1}, β_{e2} with the corresponding Gaussian distributions of the model edge ($\bar{l}_a$,l_a^σ, $\overline{\beta}_{a1}$,$\overline{\beta}_{a2}$, β_{a1}^σ,β_{a2}^σ). Identity with the mean yields a confidence of 0, the minimum confidence is -1. The confidence for the edge's appearance equals the negative distance between the edge descriptor and the model edge descriptor $\bar{\mathbf{e}}_a$. The overall confidence of edge e representing the model edge a is finally set to the minimum of the confidences for length, angles and descriptor, thus effectively filtering out unlikely candidates.

It can occur that no interest point is detected in one location of the medical structure in the target image where the model would expect one. It is thus important to include the possibility of omitting a model point. This is achieved by adding one artificial target interest point (dummy point), yielding $\mathbf{Cd}$ and $\mathbf{Ed}$ of sizes $M \times N + 1$ and $A \times (N+1)^2$, respectively. The last column of $\mathbf{Cd}$ is set to the mean of $\mathbf{C}$ multiplied by a factor f controlling how costly it should be to omit a model point. Similarly, the edges of $\mathbf{Ed}$ involving the dummy point are set to f times the mean of $\mathbf{E}$.

The MAX-SUM solver is then applied on $\mathbf{Cd}$, $\mathbf{Ed}$, yielding the set $\mathbf{S} = \{n_1, \ldots, n_M\}$ of optimal labels for each model node, maximizing the quality C in Eq. 3.

1087



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

Figure 4: (a,b) Model graph $\mathscr{A}$ automatically generated from the M selected interest points (landmarks) depicted for two of the training images. (c,d) The results of the model matching for two test images.

5 Experiments

The approach was evaluated[1] on 2 data sets (Fig. 4). **1.** For a set of 25 hand radiographs (300×450 pixels) 17 landmarks in each image were manually annotated. **2.** On 8 spine MRIs (280×320 pixels) manual annotations of the centers of 6 inter-vertebral discs were used for validation, plus 2 landmarks to disambiguate the matching. The error between found landmarks and ground truth landmarks was recorded, where only the points of medical interest (only the 6 spine landmarks which correspond to vertebral discs) where considered. The typical number of detected interest points was between 400 and 600, the model graphs contained 17 and 8 nodes, respectively. In Fig. 4 (a,b) the model graphs are depicted on two of the training images. In Fig. 4 (c,d) matching results are depicted: the red lines represent the model graph matched to the target image, while the green circles are the positions of the found landmarks.

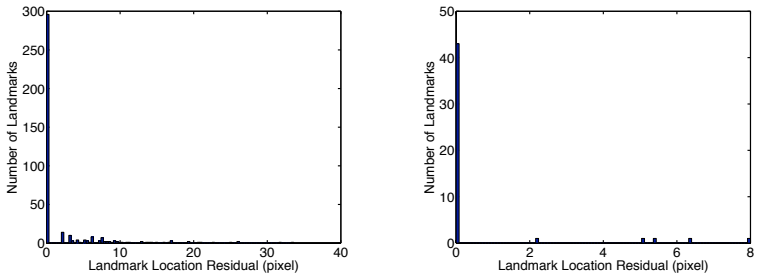
Quantitative analysis was performed by a leave-one-out procedure i.e a single image was chosen as target image and the model graph was built from the the remaining 24 or 7 images respectively. The mean/median error for matches is 2.79 / 0.0 pixels for hand data and 0.56 / 0.0 pixels for the spine data, reflecting the excellent matching accuracy. The error histograms for both sets are depicted in Fig. 5. Typical run times for solving the MRF are in the order of few seconds.

6 Conclusion and Outlook

We present a framework for the fast and accurate localisation of anatomical structures. Configurations of symmetry interest points and local descriptors derived from Gradient Vector Flow are represented by graphs and Markov Random Fields. The matching is performed by the MAX-SUM algorithm. The approach integrates local descriptor similarities and deformation constraints in a single optimization step. Results indicate that the method provides the localization accuracy necessary for the initialization of subsequent segmentation algorithms. Future research will focus on improvements to allow for the application to segmentation tasks as well as the extension to 3-dimensional data sets.

[1] The implementation used in this evaluation is available at `http://www.aamir.at/bmvc07/`

1088



(a) (b)

Figure 5: Result histograms for the pixel distances of result landmarks to ground truth landmarks for (a) the hand radiograph data set and (b) the spine MRI data set. Note the high quality of the model matching, with most of the landmarks being matched perfectly.

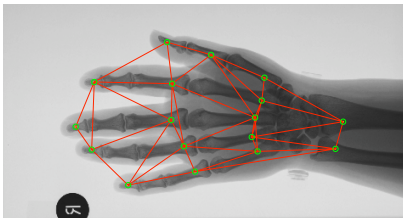


Figure 6: Example of the rotation invariance of Sparse Appearance Models: The model was trained on upright hand radiographs. As only relative angles are modeled, the hand is successfully detected in the rotated image.

References

[1] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE PAMI*, 24(4):509–522, 2002.

[2] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *Proc. ICCV*, pages 105–112, 2001.

[3] T. Cootes. Active shape models - 'smart snakes'. In *Proc. BMVC*, 1992.

[4] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Trans. PAMI*, 23(6):681–685, 2001.

[5] R. Donner, M. Reiter, G. Langs, P. Peloschek, and H. Bischof. Fast active appearance model search using canonical correlation analysis. *IEEE TPAMI*, 28(10):1690 – 1694, October 2006.

[6] M. A. Fischler and R. C. Bolles. A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24, 1981.

[7] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal on Computer Vision*, 1:321–331, 1988.

[8] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *CVPR (2)*, pages 506–513, 2004.

[9] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *PAMI*, 28(10):1568–1583, 2006.

[10] P. Kovesi. Symmetry and asymmetry from local phase. In *Proceedings of the Tenth Australian Joint Conference on Artificial Intelligence*, pages 185–190, 1997.

[11] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Solving Markov random fields using second order cone programming. In *Proc. CVPR*, pages I: 1045–1052, 2006.

[12] G. Langs, P. Peloschek, R. Donner, M. Reiter, and H. Bischof. Active feature models. In *Proc. ICPR*, pages 417–420, 2006.

[13] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[14] G. Loy and J.-O. Eklundh. Detecting symmetry and symmetric constellations of features. In *Proceedings of ECCV '06*, 2006.

[15] G. Loy and A. Zelinsky. Fast radial symmetry for detecting points of interest. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(8):959–973, 2003.

[16] S. C. Mitchell, J. G. Bosch, B. P. F. Lelieveldt, R. J. van der Geest, J. H. C. Reiber, and M. Sonka. 3-d active appearance models: Segmentation of cardiac MR and ultrasound images. *IEEE TMI*, 21(9):1167–1178, 2002.

[17] M. Roberts, T. F. Cootes, and J. E. Adams. Vertebral morphometry - semiautomatic determination of detailed shape from dual-energy x-ray absorptiometry images using active appearance models. *Investigative Radiology*, 41(12):849–459, 2006.

[18] M. Schlesinger. Sintaksicheskiy analiz dvumernykh zritelnikh signalov v usloviyakh pomekh (syntactic analysis of two-dimensional visual signals in noisy conditions). *Kibernetika*, (4):113–130, 1976. In Russian.

[19] M. B. Stegmann, H. Ólafsdóttir, and H. B. W. Larsson. Unsupervised motion-compensation of multi-slice cardiac perfusion MRI. *Medical Image Analysis*, 9(4):394–410, aug 2005.

[20] H. Thodberg and A. Rosholm. Application of the active shape model in a commercial medicals device for bone densitometry. In *BMVC*, pages 43–52, 2001.

[21] T. Werner. A linear programming approach to Max-sum problem: A review. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 29(7), 2007.

[22] C. Xu and J. L. Prince. Snakes, shapes, and gradient vector flow. *IEEE Trans. on Image Proc.*, 7(3), March 1998.

[23] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, 51(7):2282–2312, 2005.

Indoor Place Recognition using Online Independent Support Vector Machines

Francesco Orabona, Claudio Castellini
LIRA-Lab, University of Genova
Genova, Italy
{bremen,drwho}@liralab.it

Barbara Caputo, Jie Luo
IDIAP Research Institute / EPFL
Martigny, Switzerland
{bcaputo,jluo}@idiap.ch

Giulio Sandini
Italian Institute of Technology
Genova, Italy
giulio.sandini@iit.it

Abstract

In the framework of indoor mobile robotics, *place recognition* is a challenging task, where it is crucial that self-localization be enforced precisely, notwithstanding the changing conditions of illumination, objects being shifted around and/or people affecting the appearance of the scene. In this scenario online learning seems the main way out, thanks to the possibility of adapting to changes in a smart and flexible way. Nevertheless, standard machine learning approaches usually suffer when confronted with massive amounts of data and when asked to work online. Online learning requires a high training and testing speed, all the more in place recognition, where a continuous flow of data comes from one or more cameras. In this paper we follow the Support Vector Machines-based approach of Pronobis et al. [26], proposing an improvement that we call Online Independent Support Vector Machines. This technique exploits linear independence in the image feature space to incrementally keep the size of the learning machine remarkably small while retaining the accuracy of a standard machine. Since the training and testing time crucially depend on the size of the machine, this solves the above stated problems. Our experimental results prove the effectiveness of the approach.

1 Introduction

Place recognition is an open and highly challenging problem in computer vision, especially when applied to mobile robotics in indoor environments. Simply stated, the problem is that of determining what room of a house or office a mobile robot is in, based upon what the robot sees through one or more cameras. The problem is made very difficult by at least three factors: (a) the input space is huge, since we deal with images, usually at a reasonable resolution and in colour; (b) images of the same place can be quite different as illumination conditions change and moving obstacles get in the way; and (c), recognition must be done on-line in real time, as the robot is moving around. The topic is widely researched, but incremental learning approaches have been so far mostly used for

constructing the geometrical map, or the environment representation, online [6, 1]. Robustness to illumination changes, and more generally to realistic visual variations in time, has been addressed in [26], where it was shown that a pure learning approach can be very effective for tackling the first two issues: indeed it was demonstrated that an approach based upon Support Vector Machines (SVM, see, e.g., [4]) *in batch mode* could achieve a remarkable robustness to illumination changes and variability due to the normal use of the environments. At the same time the work elicited the problem of the growth of the testing time when a bigger training set was used, to have better recognition performances. In fact, as far as the third issue is concerned, it is well known that both the training and testing time of an SVM crucially depend on the number of samples considered [16]; as well, the number of Support Vectors (SVs) found, which determine the complexity of the solution to the problem, grows proportionally with respect to the number of samples [28]. This makes the approach unsuitable, at least so far, for on-line learning, where a potentially endless flow of data is acquired by the machine. SVMs can be up to 50 times slower than other specialized approaches with similar performances [8]. Several exact and approximate approaches have been proposed so far for simplifying the SVM decision function: see for instance [12], based upon linear independence of the SVs in the feature space performed after training, and other after-training simplification methods (e.g. chapter 18.3 in [27] and [23]). The exact solution to online SVM learning was given by Cauwenberghs and Poggio in 2000 [9], but their algorithm cannot be used to reduce the number of SVs. In [29] and [25] approximate incremental versions of the SVM are proposed, that also achieve a reduction of the number of SVs with small degradation of their performances.

In this paper we propose an improvement to SVMs that we call Online Independent Support Vector Machines (OISVMs). OISVMs incrementally select "basis vectors" that are used to build the solution of the SVM training problem, based upon *linear independence in the feature space*: vectors which are linearly dependent on already stored ones are rejected, and a smart, incremental minimization algorithm is employed to find the new minimum of the cost function. This keeps the number of SVs much smaller than usual, reducing the complexity of the solution and therefore both the training and testing time. Unsupervised rank reduction methods have been proposed [3] as well as supervised ones [2] that achieve the same goals, but no application of these ideas appears so far, to the best of our knowledge, in online settings. This is particularly important since in an online setting the size of a SVM would grow indefinitely, and so would the testing time. Our experiments instead indicate that the number of basis vectors of OISVMs does not grow linearly with the training set, but reaches a limit and then stops growing. This result is theoretically confirmed, e.g., in [14], even in the case the feature space is infinite-dimensional.

Such an approach is actually what is needed to tackle the problem of place recognition in mobile robotics. To support this claim, we show a set of experimental results obtained by comparing SVMs and OISVMs on a real-world place recognition problem in an indoor environment. Data images are acquired continuously by two robot platforms under different weather conditions and across a time span of several months. Our results show that our method achieves a speed-up of $3.5 - 2.3$ times with respect to the time required by the standard SVM, respectively with χ^2 kernel and matching kernel [30], while retaining essentially the same accuracy.

The paper is structured as follows: after an overview of background mathematics proper to SVMs, Section 3 describes OISVMs. Section 4 shows the experimental results

and lastly, in Section 5, conclusions are drawn and future work is outlined.

2 Background Mathematics

Due to space limitations, this is a very quick account of SVMs — the interested reader is referred to [7] for a tutorial, and to [11] for a comprehensive introduction to the subject. Assume $\{\mathbf{x}_i, y_i\}_{i=1}^{l}$, with $\mathbf{x}_i \in \mathbb{R}^m$ and $y_i \in \{-1, 1\}$, is a set of samples and labels drawn from an unknown probability distribution; we want to find a function $f(\mathbf{x})$ such that $sign(f(\mathbf{x}))$ best determines the category of any future sample $\mathbf{x}$. In the most general setting,

$$f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b \tag{1}$$

where $b \in \mathbb{R}$ and $K(\mathbf{x}_1, \mathbf{x}_2) = \Phi(\mathbf{x}_1) \cdot \Phi(\mathbf{x}_2)$, the *kernel function*, evaluates inner products between images of the samples through a non-linear mapping Φ. The α_is are Lagrangian coefficients obtained by solving (the dual Lagrangian form of) the problem

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \tfrac{1}{2}||\mathbf{w}||^2 + C\textstyle\sum_{i=1}^{l} \xi_i^p \\ \text{subject to} \quad & y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{aligned} \tag{2}$$

where $\mathbf{w}$ defines a separating hyperplane in the *feature space*, i.e., the space where Φ lives, whereas $\xi_i \in \mathbb{R}$ are slack variables, $C \in \mathbb{R}^+$ is an error penalty coefficient and p is usually 1 or 2. In practice, most of the α_i are found to be zero after training; the vectors with an associated α_i different from zero are called *support vectors*. Notice that, from (1), the testing time of a new point is proportional to the number of SVs, hence reducing the number of SVs implies reducing the testing time.

In the following, the term *kernel dimension* will refer, as is customary, to the dimension of the feature space. The kernel dimension is related to the generalization power of the machine, and it depends on the choice of the kernel itself. Widely used kernels include the *polynomial* one (finite-dimensional) and the *Gaussian* one (infinite-dimensional).

3 Online Independent Support Vector Machines

Let the *kernel matrix* K be defined such that $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$, with $i, j = 1, \ldots, l$. The possibility to obtain a more compact representation of $f(\mathbf{x})$ follows from the fact that the solution to a SVM problem (that is, the α_is) is not unique if K does not have full rank [7], which is equivalent to some of the SVs being linearly dependent on some others *in the feature space* (this is the core of Downs et al.'s [12] original idea). In an online setting, to apply Downs et al.'s idea, or any other post-training method to reduce the number of SVs, means to simplify the solution each time a new sample is acquired, which is obviously infeasible. We need a way to use independent SVs only, that is to decouple the concept of "basis" vectors, used to build the classification function (1), from the samples used to

evaluate the ξ_i in (2). If the selected basis vectors span the same subspace as the whole sample set, the solution found will be equivalent — that is, we will not lose any precision.

We hereby propose, after having received a new training sample, to incrementally add it to the basis if it is linearly independent in the feature space from those already present in the basis itself. The solution found is *the same* as in the classical SVM formulation; therefore, no approximation whatsoever is involved, unless one gives it up in order to obtain even fewer support vectors (see Section 4 for a deeper discussion on this point).

Denoting the indexes of the vectors in the current basis, after l training samples, by $\mathscr{B}$, and the new sample under judgment by $\mathbf{x}_{l+1}$, the algorithm can then be summed up as follows:

1. check whether $\mathbf{x}_{l+1}$ is linearly independent from the basis in the feature space; if it is, add it to $\mathscr{B}$; otherwise, leave $\mathscr{B}$ unchanged.
2. incrementally re-train the machine.

Hence the testing time for a new point will be $O(|\mathscr{B}|)$, as opposed to $O(l)$ in the standard approach; therefore, keeping $\mathscr{B}$ small will improve the testing time without losing any precision whatsoever.

In the following, the notations A_{IJ} and $\mathbf{v}_I$, where A is a matrix, $\mathbf{v}$ is a vector and $I,J \subset \mathbb{N}$ denote in turn the sub-matrix and the sub-vector obtained from A and $\mathbf{v}$ by taking the indexes in I and J.

3.1 Linear independence

In general, checking whether a matrix has full rank is done via some decomposition, or by looking at the eigenvalues of the matrix; but here we want to check whether a *single* vector is linearly independent from a matrix which is already known to be full-rank. Inspired by the definition of linear independence, we check how well the vector can be approximated by a linear combination of the vectors in the set [13]. Let $d_j \in \mathbb{R}$; then let

$$\Delta = \min_{\mathbf{d}} \left\| \sum_{j \in \mathscr{B}} d_j \phi(\mathbf{x}_j) - \phi(\mathbf{x}_{l+1}) \right\|^2 \tag{3}$$

If $\Delta > 0$ then $\mathbf{x}_{l+1}$ is linearly independent with respect to the basis, and $\{l+1\}$ is added to $\mathscr{B}$. In practice, we check whether $\Delta \leq \eta$ where $\eta > 0$ is a tolerance factor, and expect that larger values of η lead to worse accuracy, but also to smaller bases. As a matter of fact, if η is set at machine precision, OISVMs retain the exact accuracy of SVMs. Notice also that if the feature space has finite dimension n, then no more than n linearly independent vectors can be found, and $\mathscr{B}$ will never contain more than n vectors.

Expanding equation (3) we get

$$\Delta = \min_{\mathbf{d}} \left(\sum_{i,j \in \mathscr{B}} d_j d_i \phi(\mathbf{x}_j) \cdot \phi(\mathbf{x}_i) - 2 \sum_{j \in \mathscr{B}} d_j \phi(\mathbf{x}_j) \cdot \phi(\mathbf{x}_{l+1}) + \phi(\mathbf{x}_{l+1}) \cdot \phi(\mathbf{x}_{l+1}) \right) \tag{4}$$

that is, applying the kernel trick,

$$\Delta = \min_{\mathbf{d}} \left(\mathbf{d}^T K_{\mathscr{B}\mathscr{B}} \mathbf{d} - 2\mathbf{d}^T \mathbf{k} + K(\mathbf{x}_{l+1}, \mathbf{x}_{l+1}) \right) \tag{5}$$

where $k_i = K(\mathbf{x}_i, \mathbf{x}_{l+1})$ with $i \in \mathscr{B}$. Solving (5), that is, applying the extremum conditions with respect to $\mathbf{d}$, we obtain

$$\tilde{\mathbf{d}} = K_{\mathscr{B}\mathscr{B}}^{-1}\mathbf{k} \tag{6}$$

and, by replacing (6) in (5) once,

$$\Delta = K(\mathbf{x}_{l+1}, \mathbf{x}_{l+1}) - \mathbf{k}^T\tilde{\mathbf{d}} \tag{7}$$

Note that $K_{\mathscr{B}\mathscr{B}}$ can be safely inverted since, by incremental construction, it is full-rank. An efficient way to do it, exploiting the incremental nature of the approach, is that of updating it recursively: after the addition of a new sample, the new $K_{\mathscr{B}\mathscr{B}}^{-1}$ then becomes

$$\begin{bmatrix} & & & 0 \\ & K_{\mathscr{B}\mathscr{B}}^{-1} & & \vdots \\ & & & 0 \\ 0 & \cdots & 0 & 0 \end{bmatrix} + \frac{1}{\Delta}\begin{bmatrix} \tilde{\mathbf{d}} \\ -1 \end{bmatrix}\begin{bmatrix} \tilde{\mathbf{d}}^T & -1 \end{bmatrix} \tag{8}$$

where $\tilde{\mathbf{d}}$ and Δ are already evaluated during the test (this method matches the one used in Cauwenberghs and Poggio's incremental algorithm [9]). Thanks to this incremental evaluation, the time complexity of the linear independence check is $O(|\mathscr{B}|^2)$, as one can easily see from Equation (6).

With this method we are approximating the original kernel matrix K with another matrix $\widehat{K}$ [2]; the quality of the approximation depends on η. In fact it is possible to show that $trace(K - \widehat{K}) \leq \eta|\mathscr{B}| \leq \eta l$, where l is the number of samples acquired [14]. If we consider a normalized kernel, that is a kernel for which $K(x,x)$ is always equal to 1, we can write $trace(K - \widehat{K})/trace(K) \leq \eta$. On the other hand a bigger η means of course a smaller number of SVs, hence it controls the trade-off between accuracy and speed of the OISVM.

3.2 Training the machine

The training method largely follows Keerthi et al. [17, 16], that we have adapted for online training. The algorithm directly minimizes problem (2) as opposed to the standard way of minimizing its dual Lagrangian form, allowing to select explicitly the basis vectors to use. We set $p = 2$ in (2) and transform it to an unconstrained problem. Let $\mathscr{D} \subset \{1, \ldots, l\}$; then the unconstrained problem is

$$\min_{\beta}\left(\frac{1}{2}\beta^T K_{\mathscr{D}\mathscr{D}}\beta + \frac{1}{2}C\sum_{i=1}^{l} max(0, 1 - y_i K_{i\mathscr{D}}\beta)^2\right) \tag{9}$$

where β is the vector of the Lagrangian coefficients involved in $f(\mathbf{x})$, analogously to the α_is in the original formulation. If we set $\mathscr{D} = \mathscr{B}$, then the solution to the problem is unique since $K_{\mathscr{B}\mathscr{B}}$ is full rank by construction. Newton's method as modified by Keerthi et al. [17, 16] can then be used to solve (9) after each new sample. When the new sample $\mathbf{x}_{l+1}$ is received the method goes as follows:

1. let $\mathscr{I} = \{i : 1 - y_i o_i > 0\}$ where $o_i = K_{i\mathscr{B}}\beta$ and β is the vector of optimal coefficients with l training samples; if $\mathscr{I}$ has not changed, stop.

Figure 1: Sample images illustrating the variations in the IDOL2 database. Images in the top row show the variability introduced by changes in illumination as well as people appearing in the environment. The middle row shows the influence of people's everyday activity (first four images) as well as larger variations which happened over a time span of 6 months. Finally, the bottom row illustrates the changes in viewpoint observed for a series of images acquired one after another in 1.6 seconds.

2. otherwise, let the new β be $\beta - \gamma \mathbf{P}^{-1}\mathbf{g}$, where $\mathbf{P} = K_{\mathscr{B}\mathscr{B}} + CK_{\mathscr{B}\mathscr{I}}K_{\mathscr{B}\mathscr{I}}^T$ and $\mathbf{g} = K_{\mathscr{B}\mathscr{B}}\beta - CK_{\mathscr{B}\mathscr{I}}(\mathbf{y}_{\mathscr{I}} - \mathbf{o}_{\mathscr{I}})$.
3. go back to Step 1.

In Step 2 above, γ is set to one. In order to speed up the algorithm, we maintain an updated Cholesky decomposition of $\mathbf{P}$. It turns out that the algorithm converges in very few iterations, usually 0 to 2; the time complexity of the re-training step is $O(|\mathscr{B}|l)$, as well as its space complexity; hence, keeping $\mathscr{B}$ small will speed up the training time as well as the testing time.

4 Place Recognition via OISVMs

In this section we report the experimental evaluation of OISVMs on the place recognition scenario, where the aim is to update the model to handle variations in an indoor environment due to human activities over long time spans.

Experiments were conducted on the IDOL2 database (Image Database for rObot Localization 2, [22]), which contains 24 image sequences acquired using a perspective camera mounted on two mobile robot platforms, while moving in an indoor laboratory environment consisting of five different rooms. The sequences were acquired under various weather and illumination conditions (sunny, cloudy, and night) and across a time span of six months. Thus, this data capture natural variability that occurs in real-world environments because of both natural changes in the illumination and human activities. Fig. 1 shows some sample images from the database, illustrating the difficulty of the task. The image sequences in the database are divided as follows: for each robot platform and for each type of illumination conditions, there were four sequences recorded. Of these four sequences, the first two were acquired six months before the last two. This means that, for each robot and for every illumination condition, there are always two sequences acquired under similar conditions, and two sequences acquired under very different conditions.

This makes the database suitable for different kinds of evaluation on the adaptability of an incremental algorithm. For further details about the database see [22].

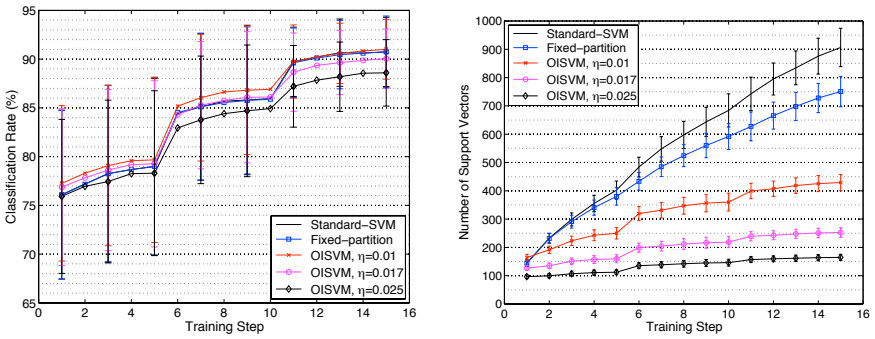
The evaluation was performed using Composed Receptive Field Histograms (CRFH) [19] as global image features and SIFT descriptors [20] of local features computed using a Harris-Laplace detector [15]. In the experiments, we consider both exponential χ^2 kernel for SVM (when use CRFH), and local kernels [30] (SIFT). Note the kernel in [30] is not always positive semidefinite [5], so this is also a test on non-Mercer kernels that have proved useful for visual recognition. The kernels used are infinite-dimensional, so for both kernels we run the OISVM using different values of η.

OISVMs have been implemented in Matlab and tested against LIBSVM v2.82 [10]. The software library has been extended to various families of kernels, and to the fixed-partition incremental SVM [29], an approximate incremental extension of SVM. In this way we can do a straightforward comparison between exact and approximate methods on this task. Notice that for the standard SVM the training is not online.

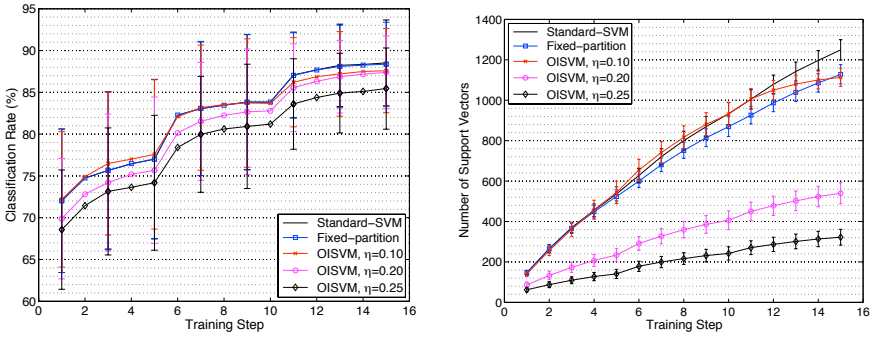
As in the experimental setup of [21], the algorithm was trained incrementally on three sequences from IDOL2, acquired under similar illumination conditions with the same robot platform; the fourth sequence was used for testing. In order to test the various properties of interest of the incremental algorithms, we need a reasonable number of incremental steps. Thus, every sequence was split into 5 subsequences, so that each subset contained one of the five images acquired by the robot every second (image sequences were acquired at a rate of 5fps). Since during acquisition the camera's viewpoint continuously changes [21], the subsequences could be considered as recorded separately in a static environment but for varying pose. This setup allows us to examine how the algorithms perform on data with less variations. In order to get a feeling of the variations of the frame images in a sequence, bottom row of Fig. 1 shows some sample images acquired within a time span of 1.6 sec. As a result, training on each sequence was performed in 5 steps, using one subsequence at a time, resulting in 15 steps in total. Overall, we considered 36 different permutations of training and test sequences for both the exponential χ^2 and matching kernels; here we report average results with standard deviation. Fig. 2, left, shows the recognition rates of the exponential χ^2 kernel (top) and matching kernel (bottom) experiments obtained at each step using OISVM, the fixed-partition algorithm and the standard SVM. Fig. 2, right, reports the number of support vectors stored in the model at each step of the incremental procedure, for both kernel types.

We see that, performance-wise, all methods achieves statistically comparable results; this is true for both kernel types. As far the machine size is concerned, the OISVM algorithm shows a considerable advantage with respect to the fixed-partition method. In the case of the exponential χ^2 kernel this advantage is truly impressive (Fig 2, top right): for $\eta = 0.017$ and 0.025 the size at the final incremental step is $34\%/22\%$ of that of the fixed-partition method and $28\%/18\%$ of that of the standard batch method. Even more important, OISVM, for these two values of η, has found a plateau in memory, while for other methods the trend seems to be of a growth proportional to the number of training data. Note that the choice of the parameter η is crucial for achieving an optimal trade-off between compactness of the solution and optimal performance.

It is very interesting to note that, in the case of the matching kernel, the memory reduction for OISVM is less pronounced, and there is not a clear plateau in memory growth by any of the algorithms. This behavior might be due to several factors: to begin with, the matching kernel is not a Mercer kernel [5], which might affect the algorithm.



(a) Number of support vectors and classification rate obtained at each incremental step using χ^2 kernel.



(b) Number of support vectors and classification rate obtained at each incremental step using local kernel.

Figure 2: Average results obtained for experiment performed on the IDOL2 database, using OISVM with three different values of η, the fixed-partition and the standard SVM.

Also, the algorithm does not reach a plateau in the SVs growth because, in the induced space of the matching kernel, there seems to be a high probability that pair of training points are orthogonal, or almost orthogonal, to each other (notice that, as the kernel is not a Mercer one, the geometric interpretation might not be valid). Anyway, given enough training points, the machine will always reach a maximum size and will stop growing [14]. Other tests on a set of standard databases commonly used in the machine learning community, as well as more details about OISVM can be found in [24].

It is worth noting that, even if the solution is kept small and the number of support vectors will be finite in any case, all the received training samples must be stored. This can be a problem in an online setting, but it could be solved using, for example, some kind of forgetting strategy. Another strategy can be the use out-of-core storage of the data (i.e., storage on the hard disk) in order to be able to deal with big training sets.

5 Discussion and conclusions

In this paper we have shown a promising improvement to Support Vector Machines, that we call Online Independent Support Vector Machines (OISVM). OISVMs can effectively

solve the problem of place recognition by a mobile robot, at least in the experiment we have shown. OISVMs were tested on the IDOL2 image database, which consists of image sequences acquired by two robot platforms under different weather conditions and across a time span of several months. OISVMs avoid using in the solution those support vectors which are linearly dependent of previous ones in the feature space. The optimization problem is solved via an incremental algorithm which benefits of the small number of the basis vectors.

As far as we know, this method is different from all analogous procedures presented so far in literature (e.g., [12, 23, 18, 31, 16]) since it is *not* an after-training simplification and it assumes *no knowledge whatsoever* of the full training set beforehand. Moreover in case of finite-dimensional kernel and $\eta = 0$, the solution is exactly the same of the standard formulation because no approximation is used.

Our experimental results show that in the case of infinite-dimensional kernels, the number of support vectors is dramatically reduced at the price of a negligible degradation in the accuracy. In fact in the case of χ^2 kernel, we get as few as 3.5 times less SVs with respect to the batch formulation and 3 times less with respect to the fixed-partition method, while retaining essentially the same accuracy. In the case of the local kernel, the speed up are respectively 2.3 and 2.1.

Since the training and testing time depend polynomially on the number of support vectors, reducing them brings an obvious speed up. A careful study of the relationship between η and the degradation in performance is being carried on; in fact, according to [14], imposing a value of η strictly larger than zero will eventually result in a *finite* number of basis vectors, *even in the case the feature space is infinite-dimensional*. Further research about finding a precise relationship between η and this number will allow us to precisely dimension the machine depending on the required precision.

Acknowledgments

This work was supported by EU projects RobotCub (IST-2004-004370), CONTACT (NEST-5010), NEUROBotics (FP6-IST-001917) and DIRAC (FP6-0027787).

References

[1] M. Artač, M. Jogan, and A. Leonardis. Mobile robot localization using an incremental eigenspace model. In *Proceedings of ICRA'02*, 2002.

[2] F. R. Bach and M. I. Jordan. Predictive low-rank decomposition for kernel methods. In *Proceedings of ICML'05*, 2005.

[3] G. Baudat and Fatiha Anouar. Feature vector selection and projection using kernels. *Neurocomputing*, 55(1-2):21–38, 2003.

[4] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *Proceedings of COLT'92*, pages 144–152. ACM press, 1992.

[5] S. Boughorbel, J.-P. Tarel, and F. Fleuret. Non-mercer kernels for svm object recognition. In *Proceedings of BMVC'04*, pages 137–146, London, England, 2004.

[6] E. Brunskill and N. Roy. Slam using incremental probabilistic pca and dimensionality reduction. In *Proceedings of ICRA'05*, 2005.

[7] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Knowledge Discovery and Data Mining*, 2(2), 1998.

[8] C. J. C. Burges and B. Schölkopf. Improving the accuracy and speed of support vector machines. In *Proceedings of NIPS'96*, pages 375–381, 1996.

[9] G. Cauwenberghs and T. Poggio. Incremental and decremental support vector machine learning. In *Proceedings of NIPS'00*, pages 409–415, 2000.

[10] 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`.

[11] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines (and Other Kernel-Based Learning Methods)*. CUP, 2000.

[12] T. Downs, K. E. Gates, and A. Masters. Exact simplification of support vectors solutions. *Journal of Machine Learning Research*, 2:293–297, 2001.

[13] Y. Engel, S. Mannor, and R. Meir. Sparse online greedy support vector regression. In *Proceedings ECML'02*, 2002.

[14] Y. Engel, S. Mannor, and R. Meir. The kernel recursive least squares algorithm. *IEEE Transactions on Signal Processing*, 52(8), 2004.

[15] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of AVS'88*, 1988.

[16] S. S. Keerthi, O. Chapelle, and D. DeCoste. Building support vector machines with reduced classifier complexity. *Journal of Machine Learning Research*, 8:1–22, 2006.

[17] S. S. Keerthi and D. DeCoste. A modified finite newton method for fast solution of large scale linear SVMs. *Journal of Machine Learning Research*, 6:341–361, 2005.

[18] Y. J. Lee and O. L. Mangasarian. RSVM: Reduced support vector machines. In *Proceedings of the SIAM International Conference on Data Mining*, 2001.

[19] O. Linde and T. Lindeberg. Object recognition using composed receptive field histograms of higher dimensionality. In *Proceedings of ICPR'04*, 2004.

[20] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of ICCV'99*, 1999.

[21] J. Luo, A. Pronobis, B. Caputo, and P. Jensfelt. Incremental learning for place recognition in dynamic environments. Accepted for *IROS'07*, to appear, 2007.

[22] J. Luo, A. Pronobis, B. Caputo, and P. Jensfelt. The KTH-IDOL2 database. Technical Report 304, KTH, CAS/CVAP, 2006. Available at `http://cogvis.nada.kth.se/IDOL2/`.

[23] D. D. Nguyen and T. B. Ho. An efficient method for simplifying support vector machines. In *Proceedings of ICML'05*, pages 617–624, New York, NY, USA, 2005. ACM Press.

[24] F. Orabona. *Learning and Adptation in Computer Vision*. PhD thesis, University of Genoa, 2007.

[25] A. Pronobis and B. Caputo. The more you learn, the less you store: memory-controlled incremental support vector machines. In *Proceedings of ICVW'06*, Gratz, 2006.

[26] A. Pronobis, B. Caputo, P. Jensfelt, and H. I. Christensen. A discriminative approach to robust visual place recognition. In *Proceedings of IROS'06*, 2006.

[27] A. Smola and B. Schölkopf. *Learning with Kernels*. MIT press, Cambridge, MA, USA, 2002.

[28] I. Steinwart. Sparseness of support vector machines. *Journal of Machine Learning Research*, 4:1071–1105, 2003.

[29] N. Syed, H. Liu, and K. Sung. Incremental learning with support vector machines. In *Proceedings of the Workshop on Support Vector Machines at IJCAI'99*, 1999.

[30] C. Wallraven, B. Caputo, and A. Graf. Recognition with local features: the kernel recipe. In *Proceedings of ICCV'03*, 2003.

[31] M. Wu, B. Schölkopf, and G. Bakir. A direct method for building sparse kernel learning algorithms. *Journal of Machine Learning Research*, 7:603–624, 04 2006.

Video-rate recognition and localization for wearable cameras

R O Castle, D J Gawley, G Klein, and D W Murray
Department of Engineering Science, University of Oxford, UK
[bob,djg,gk,dwm]@robots.ox.ac.uk

Abstract

Using simultaneous localization and mapping to determine the 3D surroundings and pose of a wearable or hand-held camera provides the geometrical foundation for several capabilities of value to an autonomous wearable vision system. The one explored here is the ability to incorporate recognized objects into the map of the surroundings and refer to them. Established methods for feature cluster recognition are used to identify and localize known planar objects, and their geometry is incorporated into the map of the surrounds using a minimalist representation. Continued measurement of these mapped objects improves both the accuracy of estimated maps and the robustness of the tracking system. In the context of wearable (or hand-held) vision, the system's ability to enhance generated maps with known objects increases the map's value to human operators, and also enables meaningful automatic annotation of the user's surroundings.

1 Introduction

Three principal threads run through research into wearable computing. The first is the creation of strata of portable and genuinely wearable hardware, and the second is the development of unobtrusive and socially acceptable sensors and interfaces to gather data and feed information back to the user. These two alone allow a degree of environmental and self monitoring by the user, or monitoring of the user by a remote operator. The last thread involves the exploration of perceptual modalities which can assess the user's environment, the user's relationship to it and activities within it, and thence augment the user's capabilities by offering contextually pertinent advice.

Work in the first thread has been greatly aided by the inexorable increase in the integration of electronic components. Wearability, however, requires account to be taken of human factors which must still be determined empirically over several cycles of design, build, and test. The hardware series from Smailagic, Sieworek and coworkers at CMU (e.g. [24]) and more recently by Tröster and colleagues at ETH (e.g. [1]) are ones where the design methodology is particularly clear. Within the second thread, sensors can be grouped into those sensing the wearer or sensing the surroundings. A raft of physiological signals has been used to determine muscle, brain and heart activity, skin conductance, respiration, blood pressure, and body temperature; and accelerometers and motion-sensitive textiles have measured user activity [17] [22] [20] [31]. Outward looking environmental sensors include those for ambient quantities like noise, or temperature (e.g. [5]); those giving just the user's position (e.g. [2]); and those giving a more fine-grained understanding of the surroundings (e.g. [27] [11] [19] [30]). Of this last group, it is visual sensing

that provides the strongest first-person perspective on the surroundings. The breadth of information available from imagery (or potentially so) makes it the "must-have" sensor for the third research thread.

Key to providing a wearable camera system with a greater degree of autonomy are the abilities both to locate the camera in the environment and to determine what is where around it [18]. In [18] it was demonstrated that once a partial 3D map of the camera's environment is established, locations can be selected for the camera to fixate upon, counteracting movements of the wearer, while continuing both to accumulate scene structure and to determine the camera pose. Whilst that process demonstrated the ability of a wearable system to direct attention independently of the wearer's movements, the 3D structure had no intrinsic significance to the wearable system. The fixated 3D points *may* have been related to known objects, but the semantic link between point and object had to be established by the wearer.

In [4] this limitation was removed by using learned appearance models to recognize objects in the scene. As well as permitting graphical augmentation of the recovered map, it was found that incorporating the recognized objects' geometry into the map improved the robustness and accuracy of localization. A minimal representation of the objects was used, which had the benefit of causing minimal disruption in the underlying localization mechanism, meaning they could run independently.

In this paper we advance the method in [4] by proposing a novel implementation with a solution to providing synchronization between the localization process, which runs regularly at video rate, and the recognition process which takes both a considerably longer and variable time. We adapt the method of delayed decision making of Leonard and Rikoski [12], a method developed to enable the initialization of features using data from multiple steps. We demonstrate the method working on a localized desk top environment, showing that a spatially-aware dialogue can be established between the wearable system and its wearer.

The following two sections briefly review the methods of establishing the camera position and map using monocular SLAM, and object identification using SIFT features. Section 4 describes the new method of organizing the combination of localization and recognition, and Section 5 gives an experimental evaluation. The paper closes with remarks on current work to combine fixation with recognition.

2 MonoSLAM

In contrast to batch methods of structure from motion recovery, simultaneous localization and mapping [26] [14] [28] places emphasis on continual recovery of the state of the camera and structure, and on maintaining information on the correlation between state members — not only to allow re-matching after neglect, but also to allow uncertainty to be reduced throughout the camera and map state vector when loop closing occurs.

Early applications were based on the extended Kalman filter [25] using landmarks in the sensor data, whether sonar [13] [14] or visual [3] [7]. The quadratic computational complexity of the EKF has made finding other methods to handle large-scale maps a major concern (e.g. [9] [10] [15] [29]), and EKF-SLAM is no longer used in its general form in field robotics. However, it remains well-suited to wearable vision using sparse landmark points. First, for wearables, sparseness of representation is no hindrance to navigation — one can rely on the wearer to get around. Secondly, the need to impose a limit on the

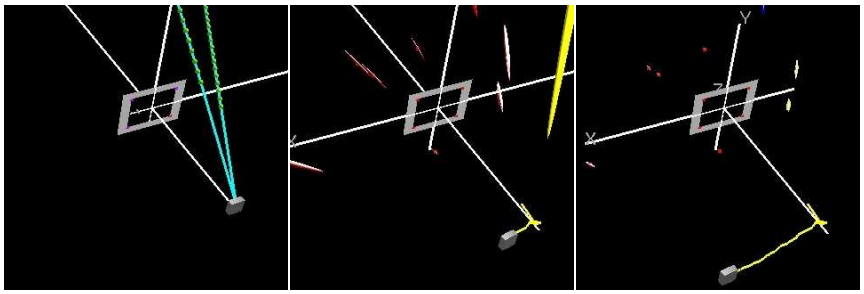


Figure 1: Typical initialization and evolution of structure and camera track in monoSLAM.

growth of the feature map in order to maintain video-rate performance is quite compatible with the notion of a local "workspace" of fixed volume around the wearer. Thirdly, such points may be annotated or recognized as points or objects of intrinsic interest to the wearer. Sparseness does make for fragility however. In monoSLAM with unconstrained camera motion, depth is not recovered from a single view or multiple views of a single point. Information comes from all points collectively, but, as processing has to be completed in a fixed time, a limit must be imposed on the feature map size.

In this paper we use the EKF monoSLAM formulation of Davison [6], [8]. The state is $\mathcal{X} = [\boldsymbol{c}, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_n]$ where the $\boldsymbol{X}$ are 3D locations of map features, and $\boldsymbol{c} = [\boldsymbol{t}, \boldsymbol{q}, \boldsymbol{v}, \boldsymbol{\omega}]$ is the camera position, orientation, translational velocity and angular velocity, all defined in the world frame. The usual non-linear state update equation, $\mathcal{X}_{k+1} = \boldsymbol{f}(\mathcal{X}_k, \boldsymbol{u}_k) + \boldsymbol{e}_k$, from time-step k to $k+1$ is assumed, where $\boldsymbol{u}_k$ is a control input, and $\boldsymbol{e}_k$ is an uncorrelated zero-mean Gaussian noise sequence. Here, as there is no source of odometry, the control input is taken to be zero. In the update, the 3D positions are assumed to be static, but the camera's state is updated according to a constant velocity model. The projections of the scene points are assumed to be related to the state at time-step k by $\boldsymbol{m}_k = \boldsymbol{h}(\mathcal{X}_k) + \boldsymbol{d}_k$ where $\boldsymbol{d}_k$ is an uncorrelated zero mean Gaussian noise sequence. The standard EKF update of the state and fully populated covariance matrix is followed.

For this implementation, "standard" [1] features for (potential) insertion into the 3D map are detected with the Shi-Tomasi saliency operator [23], and features that are eventually inserted are stored with an 11×11 pixel appearance template. Active search for correspondence is made within the predicted match region using normalized sum-of-squared difference correlation. Standard features are initialized using an inverse depth representation, using the state representation of Montiel *et al.* [21]. Figure 1 shows a typical view of recovered structure and camera track from monoSLAM that underpins the recognition process discussed below.

3 Object detection and identification

The aim now is to detect and identify known objects in the scene and to determine their location in the world frame from just a single image, while maintaining frame-rate operation. The location of a detected object will serve as an extra measurement for the SLAM

[1] The description standard merely distinguishes features used for SLAM from those used for recognition.

process. To unify recognition and localization, a point-based representation is adopted throughout, and ideally the same point features would be used for both purposes. However, Shi-Tomasi is insufficiently discriminating for recognition, making necessary a more robust method, invariant to scale and orientation changes. For this we adopt Lowe's SIFT [16], which is known to perform well, but is too computationally expensive for frame-rate operation.

3.1 The object database

The database includes at present only planar objects. To construct an entry, an image of the object is captured and, after correcting for radial distortion, SIFT descriptors $\boldsymbol{\sigma}^i$ and their positions $\boldsymbol{x}^i$, $i = 1...I$ are computed. The image need not be fronto-parallel, and so the homography $\mathtt{H}$ between the scene and image is found by choosing $n \geq 4$ image points whose corresponding scene points $\boldsymbol{X} = [X, Y, 1]^\top$ can be located in a object-based Euclidean plane. The database entry

$$\mathcal{O}_j = \{\mathcal{I}_\mathrm{R}, \{\boldsymbol{\sigma}^i, \boldsymbol{X}^i = \mathtt{H}^{-1}\boldsymbol{x}^i\}_{i=1...I}, \{\boldsymbol{X}^k_\mathrm{B}\}_{k=1...K}, \{k_1, k_2, k_3 \in 1...K\}\}_j$$

contains (i) the image $\mathcal{I}_\mathrm{R}$ of the object rectified by the homography so that it appears as a fronto-parallel view, (ii) the list of SIFT descriptors and their scene locations, (iii) the locations of several scene boundary points $\boldsymbol{X}^k_\mathrm{B}$ to define the object extent, and (iv), as explained later, the indices of three boundary points flagged for use in the SLAM map.

3.2 Object detection and localization

During a run, a video frame is selected at regular intervals and SIFT features are extracted. The detected feature locations are corrected for radial distortion[2], and are then matched to the stored keypoints of the known objects. Candidate matching descriptors are found using a pre-computed kd-tree based method [16] to search the database. If the number of matched points from any given object's database entry to the current image is greater than a threshold, we regard that object as a candidate. Because of repeated structure or other scene confusion, some of the features may be incorrectly matched. However, as the database objects are known to be planar, the database scene points $\boldsymbol{X}$ and currently observed image points $\boldsymbol{x}$ are related by a plane-to-plane homography $\boldsymbol{x} = \mathtt{H}'\boldsymbol{X}$. RANSAC is used to estimate the homography $\mathtt{H}'$ and, if a sufficiently large consensus set is found, we infer that the database object is visible in the current frame.

Having determined an object is visible we recover its location by decomposing the homography between scene and current image. In the Euclidean object-centred coordinate frame, the object lies in the plane $Z = 0$, and 3D homogeneous points on the object are $\boldsymbol{X}^{(4\times1)} = [X, Y, 0, 1]^\top$. In any view, the projection can therefore be written in terms of extrinsic and intrinsic parameters as $\boldsymbol{x} = \mathtt{K}[\mathtt{R}|\boldsymbol{t}]\boldsymbol{X}^{(4\times1)}$. Hence $\boldsymbol{x} = \mathtt{KA}\boldsymbol{X}$, where $\mathtt{A} = [\boldsymbol{r}_1\ \boldsymbol{r}_2\ \boldsymbol{t}]$ contains the translation $\boldsymbol{t}$ and the first two columns of the rotation matrix $\mathtt{R}$, all modulo a scaling factor. Using the homography already computed as the output of RANSAC and assuming known camera calibration $\mathtt{K}$, $\begin{bmatrix}\boldsymbol{r}_1 & \boldsymbol{r}_2 & \boldsymbol{t}\end{bmatrix} = \mathtt{K}^{-1}\mathtt{H}'$, again up to scale. Because the estimate $\mathtt{H}'$ is noisy, there is no guarantee that $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ found as above will be orthogonal (which they are required to be as they are columns of a rotation matrix). The closest rotation matrix, and hence the overall scale for the translation, is determined using singular value decomposition.

[2] This is faster than undistorting the whole image, and the distortion is not significant enough to effect SIFT.

The rotation matrix and translation vector calculated in this way specify the transformation of the camera from the frame of reference of an object's canonical database image. We apply this transformation in reverse to place the object in the frame of reference of the camera at the time the image was selected; and then apply a further transformation determined by the camera's pose at the time of capture relative to the world coordinate frame defined by the SLAM map to derive the position of the object in world coordinates.

3.3 Adding recognized object locations to the SLAM map

A number of methods for adding objects to the 3D map can be envisaged. The straightforward, but certainly effective, approach used here is to allow the recovered 3D position of the planar object to define 3D point measurements. The feature positions themselves are not entered, but instead the three points $\boldsymbol{X}_{\mathrm{B}}^{k}$, $k = k_1, k_2, k_3$ from the object's boundary designated in the object database entry are used. For example, for the rectangular pictures used in experiments, three of the four corners are inserted into the map. The benefits in this approach are, firstly, no additional mechanism is required in the SLAM process. Provided reasonable values are supplied for the (typically much lower) 3D error in these points, constraints on the scene will propagate properly through the covariance matrix. Secondly, there is no reliance on any particular SIFT features being re-measured over time. Thirdly, the boundary points provide a convenient representation of the extent of the object for graphical augmentation.

4 A novel implementation with delayed object insertion

The detection, localization, and SLAM methods have been re-implemented to take advantage of the capabilities of a dual core processor (2.13 GHz Intel Pentium Core 2 Duo). Including operating system overheads, monoSLAM, executing on one core with around 20 point features, takes approximately 10 ms for a 640×480 image, leaving some 20 ms per frame to perform any further computation. Object detection and localization is run in a separate thread on the second core, continuously grabbing and processing frames.

For a typical frame, SIFT detects around 500 keypoints and takes on average 700 ms to complete. Matching against a database of 16 objects containing 3.2×10^4 features takes around 100 ms. While the point based SLAM runs at 30 Hz the object detection runs at around 1.5 Hz at best. These timings will of course vary with the size of the database, the number of features found in a frame, and the number of objects found in the scene.

4.1 Delayed object insertion

Because object detection takes a variable amount of time, and because it runs much more slowly than SLAM, the process must be done in the background — that is, it must always defer to the needs of monoSLAM to run at frame-rate. A mechanism is required to permit measurement updates using recognized objects at *whatever* time the detection and recognition processes manage to complete processing a frame.

We use the delayed decision making proposed by Leonard and Rikoski [12]. Suppose the single camera SLAM system runs as normal, and that at some time step k the object recognition and object localization module described in §3.2 is able to start processing. At this point the current state vector is augmented by the camera pose, $\boldsymbol{s} = [\boldsymbol{t}, \boldsymbol{q}]$,

$$\mathcal{X}^{\mathrm{A}} = \begin{bmatrix} \boldsymbol{c} & \boldsymbol{s} & \boldsymbol{X}_1 & \cdots & \boldsymbol{X}_n \end{bmatrix}^{\top} , \tag{1}$$

No.	Object label	No. of keypoints	Image Size	Metric Size (m)
1	Colosseum	2562	480×640	0.198×0.264
2	Durdle Door	3026	600×480	0.246×0.198
3	Grasshopper	1362	600×480	0.246×0.198
⋮	⋮	⋮	⋮	⋮
14	Multiple View Geometry	1245	446×637	0.174×0.247
15	Pansy	940	600×480	0.246×0.198
16	Pots of Fire	596	480×640	0.198×0.264
	Total	31910		

Table 1: Database objects, keypoints, and the sizes.

initialized to the current pose value $\boldsymbol{s}_k$. The covariance matrix is similarly augmented

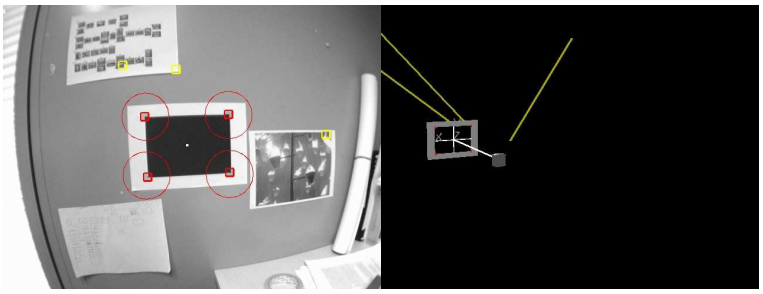
$$\mathtt{P}^{\mathrm{A}} = \begin{bmatrix} \mathtt{P}_{cc} & \mathtt{P}_{cs} & \mathtt{P}_{cX_1} & \cdots & \mathtt{P}_{cX_n} \\ \mathtt{P}_{sc} & \mathtt{P}_{ss} & \mathtt{P}_{sX_1} & \cdots & \mathtt{P}_{sX_n} \\ \mathtt{P}_{X_1c} & \mathtt{P}_{X_1s} & \mathtt{P}_{X_1X_1} & \cdots & \mathtt{P}_{X_1X_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathtt{P}_{X_nc} & \mathtt{P}_{X_ns} & \mathtt{P}_{X_nX_1} & \cdots & \mathtt{P}_{X_nX_n} \end{bmatrix}, \tag{2}$$

where $\mathtt{P}_{sc} = \mathtt{P}_{cc}[\partial \boldsymbol{s}/\partial \boldsymbol{c}]^\top$. After the saved camera pose has been added to the state, its value can no longer be directly measured. However, the correlation values contained in $\mathtt{P}^A$, between this saved pose and other elements of the state, enable its value to be updated as EKF updates continue. Therefore, as the state continues to be updated, the saved pose will be refined such that it remains consistent with newer state estimates. Once the object detection and localization completes, say n frames later, the updated saved camera pose $\boldsymbol{s}_{k+n}$ is used to determine the position of any recognized objects in the world, rather than $\boldsymbol{s}_k$. Then the saved pose is deleted from the state vector and covariance matrix. Although only one saved state is used here, the mechanism allows for multiple detection processes to start and finish at different times, were further processors available.

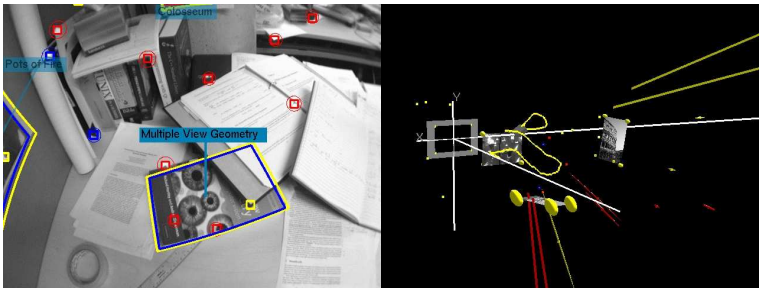
Using a saved camera pose to calculate the location of objects relies on the monoSLAM system maintaining a good estimate of the camera pose and trajectory during the intervening frames. In [4] it was shown that the inclusion of recognized object locations improved the quality of the map, and examples of object localization rescuing a failing SLAM process have been observed. However, this cannot be relied upon owing to the varying and relatively long time between object measurements. This delayed insertion method provides a faster and less complex update compared with the alternative of rolling back the EKF, inserting the measurement, and then rolling forward by recalculating all measurements from the frame the object detection was performed on.

5 Experimental evaluation

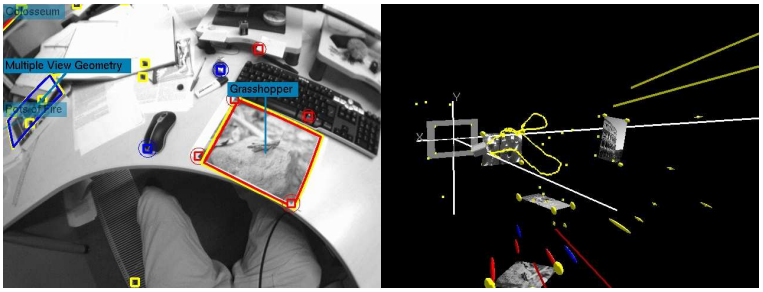
In the tests of the system reported here, a database of 16 planar objects with a total of 31,910 features was used (a sample of which is shown in Table 1), but only a subset of these objects appear in the scene. The database was created by running SIFT on each object image to generate the keypoints and measuring the metric sizes of the objects.



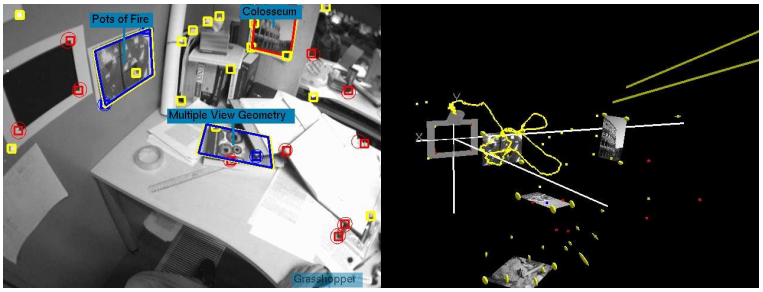
(i) Start of the sequence with the calibration plate visible in the map.



(ii) Two objects already initialized (Pots of Fire and Colosseum) and Multiple View Geometry just initialized.

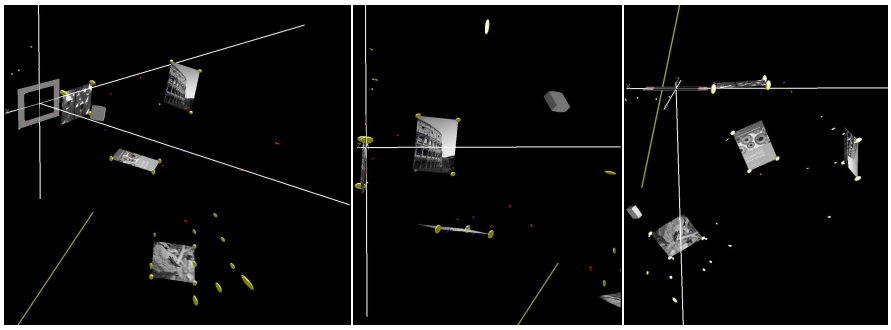


(iii) Final object (Grasshopper) located.



(iv) All objects have been detected and successfully localized.

Figure 2: The sequence runs from top to bottom with the camera view shown on the left and the map on the right.



(a) Perspective view (b) Along the x-axis (c) Along the y-axis

Figure 3: (a) View of the whole 3D map. (b,c) Individually recognized and located planar objects on the XY wall are recovered as coplanar to within map error. See Table 2.

Object label	Actual angle (°)	Measured angle (°)	Error (°)
Colosseum	90	91.2	±6
Grasshopper	90	84.7	±3
Multiple View Geometry	90	87.1	±3
Pots of Fire	0	5.0	±9

Table 2: Angles between the calibration plate and the objects.

Fig. 2 shows the evolution of processing, from initial calibration of the SLAM system to a time when there are four recognized planar objects in the SLAM map. The 2D views show the automatically generated overlaid identities and extents of the objects, typical of that which would be useful to the user of a wearable or hand-held camera. The views on the right show the evolution of the 3D map with recognized objects represented by their database image.

Fig. 3 shows various views around a particular 3D map in which there are four picture objects, one of which (Pots of Fire) should be coplanar with the calibration plate (and hence in the XY-plane), two of which (Multiple View Geometry, Grasshopper) are in the XZ plane and the final object (Colosseum) is in the ZY plane. It can be seen that all of the objects are in their respective planes to within experimental error. Table 2 shows the angles between the planes recovered from the SLAM map. Tuning the performance to the size of the covariance suggest that the lateral and depth errors are of order 10 mm and 20 mm respectively.

6 Discussion

This paper has described a system able to detect and recognize planar objects using appearance-based methods and to insert both their geometry and identity into a map — a map which is initialized and updated by an underlying monocular SLAM process which runs at fixed frame-rate using for the most part more cheaply-computed features. In particular here, the variable and relatively slow rate of delivery of geometry from the recognition process has been properly accommodated in SLAM's statistical framework using

Leonard and Rikoski's method of delayed decision-making, which inserts a temporary "place-holder" location in the state and covariance. This is updated during the time the recognition takes to complete, and is then deleted once it has been used to calculate the geometry of recognized objects. The paper demonstrates the system working in a desk top environment, providing automatic feedback on location and identity to the user.

Two avenues of application are being explored, one in the area of hand-held cameras, the second using an active wearable camera. With input from a hand-held camera, the system has no direct control over what imagery is captured. We are exploring guiding the user to different parts of the scene to search for new or already discovered objects using directional feedback provided on screen and by auditory instruction. Street frontages and art galleries are areas where the use of planarity is not a particular constraint to experimentation. When an active wearable camera supplies the imagery, the system has some autonomy to explore the world itself. As mentioned in the introduction, in [18] 3D point positions in the map were hand-labelled to allow a remote operator to command an active wearable to fixate on objects of interest while continuing to map. This method can now be automated to command the system to locate and fixate upon particular objects, without intervention of the wearer. Another avenue of exploration is that of extending the method to non-planar objects. There seems no fundamental impediment to doing so.

7 Acknowledgements

This work was supported by UK Engineering and Physical Science Research Council (grants GR/S97774 and EP/D037077). The authors are grateful to David Lowe for the SIFT source code, and for insightful conversations with members of the Active Vision Laboratory.

References

[1] U. Anliker, J. Beutel, M. Dyer, R. Enzler, P. Lukowicz, L. Thiele, and G. Tröster. A systematic approach to the design of distributed wearable systems. *IEEE Trans on Computers*, 53(8):1017–1033, 2004.

[2] H. Aoki, B. Schiele, and A. Pentland. Realtime personal positioning system for a wearable computers. In *Proc 3rd IEEE Int Symp on Wearable Computing, San Francisco CA, Oct 18-19, 1999*, pages 37–43, 1999.

[3] N. Ayache and O.D. Faugeras. Maintaining representations of the environment of a mobile robot. *IEEE Transactions on Robotics and Automation*, 5(6):804–819, 1989.

[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 on Robotics and Automation, Rome, Italy, April 10-14, 2007*, pages 4102–4107, 2007.

[5] B. Clarkson and A. Pentland. Unsupervised clustering of ambulatory audio and video. In *Proc IEEE Int Conf on Acoustics, Speech and Signal Processing, Phoenix AZ*, volume 6, pages 3037–3040, 1999.

[6] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proc 9th Int Conf on Computer Vision, Nice France, Oct 13-16, 2003*, 2003.

[7] A.J. Davison and D.W. Murray. Mobile robot localisation using active vision. In *Proc 5th European Conf on Computer Vision, Freiburg, Germany, May*, pages 809–825. Springer-Verlag, 1998.

[8] A.J. Davison, I.D. Reid, N. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, accepted for publication, 2007.

[9] M.W.M.G. Dissanayake, P.M. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, 2001.

[10] Joss Knight, Andrew Davison, and Ian Reid. Towards constant time SLAM using postponement. In *Proc. IEEE/RSJ Conf. on Intelligent Robots and Systems, Maui, HI*, volume 1, pages 406–412. IEEE Computer Society Press, October 2001.

[11] M. Kourogi, T. Kurata, and K. Sakaue. A panorama-based method of personal positioning and orientation and its real-time applications for wearable computers. In *Proc 5th IEEE Int Symp on Wearable Computing, Oct 2001*, pages 107–114, 2001.

[12] J. Leonard and R. Rikoski. Incorporation of delayed decision making into stochastic mapping. In *D. Rus and S. Singh, editors, Experimental Robotics VII, Lecture Notes in Control and Information Sciences. SpringerVerlag*, 2001.

[13] J.J. Leonard and H.F. Durrant-Whyte. *Directed Sonar Sensing for Mobile Robot Navigation*. Kluwer Academic, Boston MA, 1992.

[14] J.J. Leonard, H.F. Durrant-Whyte, and I.J. Cox. Dynamic map building for an autonomous mobile robot. *International Journal of Robotics Research*, 11(8):286–298, 1992.

[15] J.J. Leonard and P.M. Newman. Consistent, convergent and constant-time SLAM. In *Int Joint Conference on Artificial Intelligence, 2003*, volume 18, pages 1143–1150. Morgan Kaufmann, 2003.

[16] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[17] S. Mann, D. Chen, and S. Saman. HI-Cam: intelligent biofeedback signal processing. In *Proc 5th IEEE Int Symp on Wearable Computing, Oct 2001*, pages 178–179, 2001.

[18] W. W. Mayol, A. J. Davison, B. J. Tordoff, and D. W. Murray. Applying active vision and slam to wearables. In P. Dario and R. Chatila, editors, *International Symposium on Robotics Research, Siena, Italy, October 19-21, 2003*, volume 15, pages 325–334. Springer, 2003.

[19] W.W. Mayol, B.J. Tordoff, and D.W. Murray. Wearabale visual robots. *Personal and Ubiquitous Computing*, 6:37–48, 2002.

[20] J. Meyer, P. Lukowicz, and G. Tröster. Textile pressure sensor for muscle activity and motion detection. In *Proc 10th IEEE Int Symp on Wearable Computers, Montreux, Switzerland, Oct 11-14, 2006*, 2006.

[21] J. M. M. Montiel, J. Civera, and A. J. Davison. Unified inverse depth parametrization for monocular SLAM. In *Proc Conf on Robotics: Science and Systems, Philadelphia PA, Aug 16-19*, 2006.

[22] D. Raskovic, T. Martin, and E. Jovanov. Medical monitoring applications for wearable computing. *The Computer Journal*, 47(4):495–504, 2004.

[23] J. Shi and C. Tomasi. Good features to track. In *Proc IEEE Conf on Computer Vision and Pattern Recognition, Seattle WA, June 21-23, 1994*, pages 593–600, 1994.

[24] A. Smailagic and D. Siewiorek. System level design as applied to CMU wearable computers. *Journal of VLSI Signal Processing Systems*, 21(3), 1999.

[25] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In I. Cox and G. Wilfong, editors, *Autonomous Robot vehicles*, pages 167–193. Springer-Verlag New York, Inc., New York, NY, USA, 1990.

[26] R.C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56–68, 1986.

[27] T. Starner, B. Schiele, and A. Pentland. Visual contextual awareness in wearable computing. In *Proc 2nd IEEE Int Symp on Wearable Computing, Pittsburgh PA, Oct 19-20, 1998*, pages 50–57, 1998.

[28] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, Cambridge MA, 2005.

[29] S. Thrun, Y. Liu, D. Koller, A. Ng, Z. Ghahramani, and H. Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *International Journal of Robotics Research*, 23(7-8):693–716, 2004.

[30] J.A. Ward, P. Lukowicz, G. Tröster, and T.E. Starner. Activity recognition of assembly tasks using body-worn microphones and accelerometers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1553–1567, 2006.

[31] T. Westeyn, P. Presti, and T. Starner. ActionGSR: A combination galvanic skin response-accelerometer for physiological measurements in active environments. In *Proc 10th IEEE Int Symp on Wearable Computers, Montreux, Switzerland, Oct 11-14, 2006*, pages 129–130, 2006.

Probabilistic egomotion from a statistical framework

Hitesh Shah and Arvind Lakshmikumar
Sarnoff Innovative Technologies Pvt. Ltd.
Bangalore, India
(hshah, alakshmikumar)@sarnoff.com

Abstract

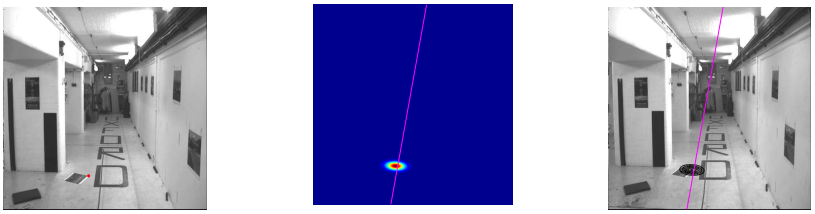
Traditional egomotion estimation algorithms have largely depended on deterministic feature correspondences to infer information about the camera and have been oblivious to the scene geometry by treating scenes with varying projectivities uniformly. This paper builds on the statistical framework of the joint feature distribution (JFD) which models the joint probability distributions of the positions of corresponding features in different images. This framework explicitly gives probabilistic correspondence search regions that can be stably estimated for the whole range of planar, shallow and deep scenes using relatively few correspondences. These joint probability distributions are constrained by the epipolar constraint to yield a distribution over all feasible egomotions. The paper also compares the proposed method against existing well-known methods and quantifies the improvements in the egomotion estimates.

1 Introduction

Egomotion estimation is a critical step while analyzing scenes from moving cameras. The aim of egomotion computation is to estimate translation and rotation, i.e. external camera parameters, the camera undergoes while capturing the sequence of images. An array of methods to estimate egomotion of moving cameras with respect to both stationary and dynamic scenes using a deterministic framework have been proposed. Tian et. al [15] and Armangue et. al [1] summarize these methods and group them based on their underlying principles.

Kanatani's method [7] based on the epipolar constraint has been the basis for several linear egomotion algorithms. The Essential matrix which serves as the support for the epipolar constraint faithfully captures the epipolar geometry between the camera views and can be estimated using the linear 8 point algorithm [5], or the state of the art 5 point algorithm [11] [9].

The above mentioned methods directly use feature correspondences between images and use these matches to robustly estimate the Essential matrix. However, given the vagaries in scene structure, extracting dense feature correspondences between images is not always possible. This paper builds on the premise that robust egomotion estimation does not have to rely on dense, deterministic image correspondences. Instead, a probability distribution of the uncertainties in correspondences would be sufficient. This paper uses the joint feature distributions (JFD) [17] to build these probability distributions. The JFD



(a) (b) (c)

Figure 1: Joint Feature Distribution (JFD) and Epipolar line : (a) shows the point under consideration marked in red. (b) shows the corresponding epipolar line (calculated using the ground truth data) and probability distribution of the point correspondence using JFD, (c) the iso-contour plot of the probability distribution and the epipolar line overlaid on the second image.

is used to predict feature correspondences between images. Probability distribution for the Essential matrix is computed from the JFD by evaluating how well each tentative Essential matrix's epipolar lines fit the feature correspondence distribution.

The motivating application for this paper is to estimate the motion of a vehicle using a rigidly attached camera. In some cases this task becomes difficult as multiple hypotheses may fit the epipolar geometry for e.g. the camera view may be of shallow scene or a deep scene or a largely planar scene. If an incorrect hypothesis is chosen, the estimated motion can break and lead to an incorrect state from which the motion estimate is unlikely to recover. The uncertainty estimates from the joint feature distributions provide us with a mechanism to choose the best hypothesis from the space of available choices.

The paper is organized as follows. Section 2 motivates the requirement to represent the correspondence information probabilistically and later describes our chosen method of probabilistic representation. Section 3 goes on to describe the method used to extract egomotion information from this probabilistic representation. Section 4 compares and contrasts our approach with other known methods in literature. The paper ends with a summary on future work.

2 Probabilistic Correspondence

Estimation of the correspondences of features between images is a difficult task. Traditionally, a feature detector (e.g. the Harris corner detector [4]) is used to find points whose correspondence is most easily established. Then, matching techniques are used to find probable matches between the feature points in both images (e.g. normalized cross correlation, or SIFT features [10]).

Most feature extraction and subsequent matching process are hindered by noise, scale, orientation changes, aperture effect, and repetitive scene structure. The ambiguities arising from these effects would cause feature based matching techniques to reject true feature points as weak ones or as outliers. A probability distribution gives us a mechanism for representing this ambiguity.

There is a good deal of literature regarding representing these ambiguities explicitly using probabilistic methods. For the purpose of computing optical flow [2] estimates traditional flow vectors at each point, by first estimating flow probability distributions, and then combining this information using spatiotemporal support regions. For the same

problem [14] creates a probability distribution over the optical flow by assuming image gradients are corrupted by a Gaussian noise model. These distributions are then used to estimate optical flow vectors with higher accuracy. Object tracking has also been addressed [13] using the probabilistic notions of correspondence. In [3] the authors proposes a method to compute correspondence probability distributions using Gabor filters that are tuned to different orientations and scales. They use the fact that for a given filter, matching points will have matching phase. They further illustrate the application of this approach to the problem of egomotion estimation. However, their method is only suitable for situations with limited rotation or scale change and does not have a mechanism to counter the effects of varying projectivities in a scene. Also, in presence of regularly repetitive texture, the responses of the Gabor filter bank are identical at multiple places and this would lead to problems during the egomotion estimation phase.

Joint Feature distribution (JFD) [17] allows statistical representation of feature correspondences in different image as a probability distribution. The probability distribution serves to capture the correspondences entirely as conditioning on a feature gives tight probabilistic correspondence search regions for the remaining ones. As concisely stated in [17], JFDs are descriptive statistical models rather than normative geometric ones: they aim to summarize the observed behavior of the given training correspondences, not to rigidly constrain them to an ideal predefined geometry.

The problem we address in this paper is egomotion estimation of a camera mounted on a moving vehicle. This application involves situations with deep scale and steep rotation changes. The JFD's offer a principled mechanism to generate the joint distributions of feature points that undergo these changes. The uncertainty distributions generated by the JFD are then used as the input to our egomotion algorithm.

2.1 Joint Feature Distributions

Noisy image projections $x_i|i = 1 \cdots m$ of a fixed 3D feature f can be modeled as probability distributions $p(x_i|f)$ centered on $f's$ true projections. If f varies across some 3D features with distribution $p(f)$, the joint feature distribution (JFD) of the resulting population of image features is

$$\mathbf{p}(x_1, \cdots, x_m) = \int p(x_1, \cdots, x_m|f)p(f)df. \tag{1}$$

JFDs are image-based models originally derived from 3D quantities (in this case, the 3D feature prior $p(f)$ and the projection models $p(x_i|f)$), but typically estimated from observed image correspondences.

Let $\mathbf{x} = (x, y, 1)$ and $\mathbf{x}' = (x', y', 1)$ be the homogeneous coordinates of the corresponding points (correspondences established using traditional feature detection and matching) in the image im_1 and im_2. Then a joint image vector is defined as

$$\mathbf{t} = \mathbf{x} \otimes \mathbf{x}' = (xx', xy', x, yx', yy', y, x', y', 1). \tag{2}$$

Given N correspondences between images $im1$ and $im2$ a 9 x N matrix $\mathbf{M}$ is obtained by stacking the joint image vectors. Thus $\mathbf{M} = [\mathbf{t_1}\ \ \mathbf{t_2}\ \ \ldots\ \ \mathbf{t_N}]$. The homogeneous scatter matrix $\mathbf{V}$ is

$$\mathbf{V} = \frac{1}{N}\sum_p \mathbf{t}_p\mathbf{t}_p^T = \frac{1}{N}\mathbf{M}\mathbf{M}^T.$$

The fundamental matrix uses just the smallest eigenvector of $\mathbf{M}\mathbf{M}^T$ whereas the JFD model captures the underlying uncertainty using an appropriately-weighted average over all of the eigenvectors ($(\mathbf{M}\mathbf{M}^T)^{-1}$). Conditioning the JFD gives compact correspondence search regions consistent with all the likely models in the average. The JFD information matrix that forms the basis of our probabilistic representation is $\mathbf{W} \approx \mathbf{V}^{-1}$. Now the probability of a point $\mathbf{x} = (x, y, 1)$ to correspond to $\mathbf{x}' = (x', y', 1)$ using JFD is given by

$$\mathbf{p}(\mathbf{x}, \mathbf{x}') = \mathbf{k_i} \mathbf{e}^{\frac{-1}{2}\mathbf{t}^T \mathbf{W} \mathbf{t}}, \tag{3}$$

where $\mathbf{t}$ is as defined by equation (2) and $\mathbf{k_i}$ is a constant to normalize the distribution.

2.2 Reducing the probability distribution

The joint image vector can be reformulated as follows

$$\begin{aligned} \mathbf{t} &= \mathbf{x} \otimes \mathbf{x}' \\ &= [xx', xy', x, yx', yy', y, x', y', 1]^T_{9\times 1} \\ &= \begin{bmatrix} xI_3 \\ yI_3 \\ I_3 \end{bmatrix}_{9\times 3} \bullet \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}_{3\times 1} \end{aligned}$$

where I_3 is a 3×3 identity matrix. Let $\mathbf{Q} = [xI_3 \quad yI_3 \quad I_3]^T$, thus $\mathbf{t} = \mathbf{Q} \cdot \mathbf{x}'$. Using equation (3), the probability for correspondence between $\mathbf{x}$ and $\mathbf{x}'$ is given by

$$\begin{aligned} \mathbf{p}(\mathbf{x}, \mathbf{x}') &= \mathbf{k_i} \mathbf{e}^{\frac{-1}{2}\mathbf{x'^T}\mathbf{Q}^\mathbf{T}\mathbf{W}\mathbf{Q}\mathbf{x}'} \\ &= \mathbf{k_i} \mathbf{e}^{\frac{-1}{2}\mathbf{x'^T}\mathbf{A}\mathbf{x}'} \end{aligned} \tag{4}$$

where $\mathbf{A} = \mathbf{Q}^\mathbf{T}\mathbf{W}\mathbf{Q}$.

Figure 1 shows the probability distribution obtained using the equation (4). It can be noted that the probability distribution models the correspondence and the associated uncertainty well.

3 Egomotion Estimation

Egomotion of a moving camera is in essence the relative geometry between subsequent camera views. This geometry is well captured by the 3×3 homogeneous Essential matrix. Consider a camera with constant intrinsic matrix $\mathbf{K}$ observing a static scene. Two corresponding image points $\mathbf{x}$ and $\mathbf{x}'$are then related by a fundamental matrix $\mathbf{F}$:

$$\mathbf{x'^T}\mathbf{F}\mathbf{x} = \mathbf{0} \tag{5}$$

A valid $\mathbf{F}$ must also satisfy the cubic singularity condition $\det \mathbf{F} = 0$. If the camera is fully-calibrated with $\mathbf{K}$ as the internal camera calibration matrix, then the fundamental matrix is reduced to an essential matrix denoted by $\mathbf{E}$, and the new equation is:

$$\mathbf{K}^{-\mathbf{T}}\mathbf{E}\mathbf{K}^{-\mathbf{1}} = \mathbf{F}. \tag{6}$$

The Essential matrix $\mathbf{E}$ is a representation of the motion (translation and rotation, up to a scale), it has only five degrees of freedoms. Consequently, to be a valid essential matrix $\mathbf{E}$, it must further satisfy two more constraints, which are characterized by

$$2\mathbf{E}\mathbf{E}^{\mathbf{T}}\mathbf{E} - Tr(\mathbf{E}\mathbf{E}^{\mathbf{T}})\mathbf{E} = \mathbf{0}, \tag{7}$$

where $Tr(\mathbf{A})$ is the trace of the matrix $\mathbf{A}$. The above constraints can also be satisfied by formulating the Essential matrix in terms of the translation and rotation the camera undergoes. A unit length vector for translation can be uniquely represented by a point on the unit sphere. Thus it can be characterized by two parameters (α, β).

$$\mathbf{T} = [\sin(\alpha)\cos(\beta), \sin(\alpha)\sin(\beta), \cos(\alpha)]^T$$

The rotation is represented by a vector $\omega = [x, y, z]^T$. Here the angle of rotation is $\theta = \sqrt{x^2 + y^2 + z^2}$ and the axis of rotation is $\hat{\omega} = \omega/\theta = [\hat{x}, \hat{y}, \hat{z}]^T$. Thus given the 5 parameters (α, β, x, y, z) the essential matrix can be composed as follows.

$$\mathbf{E} = [\mathbf{T}]_{\times}\mathbf{R}(\omega), \tag{8}$$

where $\mathbf{R}(\omega)$ is the rotation matrix corresponding to the rotation vector ω.

3.1 Probability of egomotion given a point

Given a correspondence probability distribution for a single point $\mathbf{x}$, the probability of a given hypothesis (α, β, x, y, z), and hence $\mathbf{E}$ (by equation (8)), is taken to be maximum probability $\mathbf{p}(\mathbf{x}, \mathbf{x}')$ such that $\mathbf{x}$ and $\mathbf{x}'$ satisfy the epipolar constraint, i.e. $\mathbf{x}^{\mathbf{T}}\mathbf{F}\mathbf{x}' = 0$ where $\mathbf{F}$ is the fundamental matrix given by equation (6). This translates to

$$\begin{aligned} \mathbf{P}_{\mathbf{x}}(\mathbf{E}) &= \mathbf{max}_{\mathbf{x}'} \ \mathbf{p}(\mathbf{x}, \mathbf{x}') \\ &\quad \text{subject to } \mathbf{x}^{\mathbf{T}}\mathbf{F}\mathbf{x}' = \mathbf{0} \end{aligned} \tag{9}$$

All $\mathbf{x}'$ which satisfy the epipolar constraint lie on the line given by $\mathbf{l} = \mathbf{F}\mathbf{x}$. Consider two points on the epipolar line $\mathbf{l} = [l_1, l_2, l_3]$.

$$p1 = [0, \frac{-l_3}{l_2}, 1] \quad p2 = [\frac{-l_3}{l_1}, 0, 1]$$

Any point on the epipolar line can thus be represented as

$$\mathbf{x}'(\mathbf{t}) = o + \mathbf{t}d, \tag{10}$$

where $o = p2$ and $d = (p1 - p2)$.

Equation (9) can be reformulated using equation (10) along with equation (4) to have new parameterization of $\mathbf{t}$ which inherently incorporates the epipolar constraint. This essentially converts the constrained maximization over $\mathbf{x}'$ to an unconstrained maximization over $\mathbf{t}$.

$$\begin{aligned} \mathbf{P}_{\mathbf{x}}(\mathbf{E}) &= \mathbf{max}_{\mathbf{t}} \ \mathbf{p}(\mathbf{x}, \mathbf{x}'(\mathbf{t})) \\ &= \mathbf{max}_{\mathbf{t}} \ \mathbf{k_i} e^{\frac{-1}{2}\mathbf{x}'(\mathbf{t})^{\mathbf{T}}\mathbf{A}\mathbf{x}'(\mathbf{t})} \end{aligned} \tag{11}$$

Maxima for the equation (11) will occur for a value of $\mathbf{t}$ which minimizes $\mathbf{x}'(\mathbf{t})^{\mathbf{T}}\mathbf{A}\mathbf{x}'(\mathbf{t})$. Thus

$$\tilde{\mathbf{t}} \quad = \quad \mathbf{arg}\ \mathbf{min_t}\ \ (o+\mathbf{t}d)^T A(o+\mathbf{t}d). \tag{12}$$

Minimizing the equation (12) we have $\tilde{\mathbf{t}} = -\frac{o^T\mathbf{A}d}{d^T\mathbf{A}d}$. Hence

$$\mathbf{P_x}(\mathbf{E}) \quad = \quad \mathbf{k_i} e^{\frac{-1}{2}\check{\mathbf{x}}^{\mathbf{T}}\mathbf{A}\check{\mathbf{x}}}, \tag{13}$$

where $\check{\mathbf{x}} = o - \frac{o^T\mathbf{A}d}{d^T\mathbf{A}d}d$.

3.2 Probability distribution of egomotion

The probability of egomotion computed over all the points is given by combining the information given by all points. Hence

$$\mathbf{P}(\mathbf{E}) = \mathbf{C} \prod_{\mathbf{i}}^{\forall\ \mathbf{points}} \mathbf{P_{x_i}}(\mathbf{E}), \tag{14}$$

where $\mathbf{C}$ is the normalizing constant for the distribution. Ideally we would like to consider all the points on the image, not only the points for which the correspondence was established during the initial JFD calculation phase in the above equation. However, empirically we have found that taking even few equally spaced points on a grid results in an accurate probability distribution of $\mathbf{E}$ and calculated egomotion (refer section 4).

Using equation (13) the above equation becomes

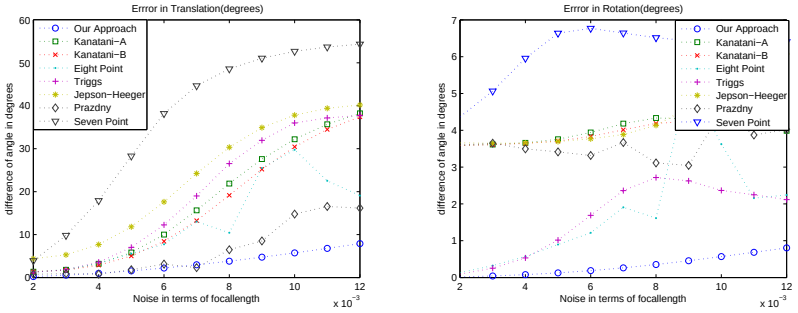
$$\mathbf{P}(\mathbf{E}) = \mathbf{C} \prod_{\mathbf{i}}^{\forall\ \mathbf{points}} \mathbf{k_i} e^{\frac{-1}{2}\check{\mathbf{x}}_{\mathbf{i}}^{\mathbf{T}}\mathbf{A}\check{\mathbf{x}}_{\mathbf{i}}}.$$

To estimate the egomotion $\tilde{\mathbf{E}}$, we find the motion parameters (α, β, x, y, z) which maximize the $\mathbf{P}(\mathbf{E})$. Hence

$$\begin{aligned}
\tilde{\mathbf{E}} \quad &= \quad \mathbf{arg}\ \mathbf{max_E}\ (\mathbf{C} \prod_{\mathbf{i}}^{\forall\ \mathbf{points}} \mathbf{k_i} e^{\frac{-1}{2}\check{\mathbf{x}}_{\mathbf{i}}^{\mathbf{T}}\mathbf{A}\check{\mathbf{x}}_{\mathbf{i}}}) \\
&= \quad \mathbf{arg}\ \mathbf{max_E}\ (\prod_{\mathbf{i}}^{\forall\ \mathbf{points}} e^{\frac{-1}{2}\check{\mathbf{x}}_{\mathbf{i}}^{\mathbf{T}}\mathbf{A}\check{\mathbf{x}}_{\mathbf{i}}}) \\
&= \quad \mathbf{arg}\ \mathbf{min_E}\ (\sum_{\mathbf{i}}^{\forall\ \mathbf{points}} \check{\mathbf{x}}_{\mathbf{i}}^{\mathbf{T}}\mathbf{A}\check{\mathbf{x}}_{\mathbf{i}})
\end{aligned} \tag{15}$$

In practice the optimization of the equation (15) over this 5 dimensional space is carried out as follows.

- Evaluate $\mathbf{P}(\mathbf{E})$ at 250 random samples in 5D motion space.
- Sort in descending order and select top 50 samples.
- Use them as seed points to start nonlinear search for optimal parameter set.
- Select the parameter set which gives minimum value for $\mathbf{P}(\mathbf{E})$ from the resulting parameter set of above nonlinear search.



(a) (b)

Figure 2: Comparison of (a) error in the translational and (b) error in rotational component of the estimated egomotion using various approaches with varying amounts of noise levels in the correspondences.

4 Experimental results

We compare the performance of the proposed approach with several well established methods on real as well as synthetic data . The error metric for estimated translation $\mathbf{T}$ with respect to the ground truth translation $\tilde{\mathbf{T}}$ is computed as

$$e_T = \cos^{-1}(\mathbf{T}'\tilde{\mathbf{T}}).$$

Similarly, the error metric for estimated rotation matrix $\mathbf{R}$ with respect to the ground truth rotation matrix $\tilde{\mathbf{R}}$ is computed as

$$e_R = \cos^{-1}\left(\frac{Tr(\mathbf{R}'\tilde{\mathbf{R}}) - 1}{2}\right).$$

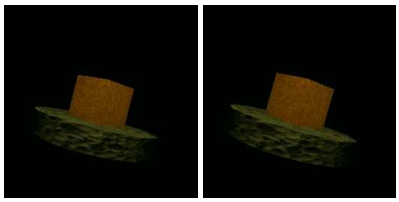
Since the internal camera parameters are assumed to be known in all the experiments on synthetic data, we have used normalized coordinates for the correspondences.

In general for minimization in 5D space it is hard to guarantee of convergence. However, due to the Gaussian nature of correspondence and parameterization of equation (15) very few local minima were observed. This coupled with evenly distributed multiple seed points in the 5D space resulted in convergence to the global minima each time in our experiments.

4.1 Synthetic data

For synthetic data, 100 3D points were randomly selected in front of the camera covering the field of view. These points were then projected on a image considering a unit focal length camera at canonical position on origin. The camera undergoes $\tilde{\mathbf{T}}$ translation and $\tilde{\mathbf{R}}$ rotation and the points are re-projected, using the new camera position, on the image. The correspondences thus generated are then perturbed by zero mean Gaussian noise to quantify the performance of various algorithms with increasing levels of noise. In our experiments, the noise variance was varied from 2×10^{-3} to 12×10^{-3} (in units of focal length) which approximately translates to 0.5 pixels to 2.5 pixels for a 512×512 image with unit focal length and 90° field of view.

Figure 2 shows the performance of the proposed approach on noisy data in comparison with Kanatani-A [7], Kanatani-B [8], Jepson-Heeger [6], Prazdny [12], Triggs [16], Eight



(a) (b)

Approach	Error in translation	Error in rotation
Proposed approach	1.0856	0.0026
Domke's approach	1.1786	0.3009
Eight point approach	2.9182	1.0144

(c)

Figure 3: Translation and rotational error in degrees for the estimated egomotion between cameras of the image (a) and (b). The comparision table is shown in (c).

Point [5], and Seven Point [18] approaches. Implementation of the Kanatani-A, Kanatani-B, Jepson-Heeger, Prazdny have been adapted from the MATLAB toolbox given by Tian et. al [15]. The Triggs method is based on the projective factorization approach proposed by [16] to calculate the projection matrices for the two cameras. These are then decomposed to obtain the egomotion. On similar lines, the Eight point and Seven point algorithms are used to obtain the Fundamental matrix. Since the image coordinates are normalized in our case, we compute the Essential matrix which is then factorized to obtain the solution for the egomotion. It can be observed from the comparisons in Figure 2 that the egomotion estimates using the proposed probabilistic approach performs better then deterministic approaches which can be attributed to the ability of the probability distribution to handle noise in the correspondences.

4.2 Synthetic and Real Images

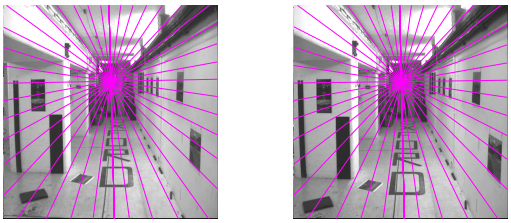
We have used synthetic image from SOFA [1] for evaluating the performance of the proposed method. The pairs of images in which the camera undergoes translation and rotation have been selected for this set of experiments.

Besides the comparisons in Figure 2, we also compare our results with the only other method, Domke et. al [3] that uses probabilistic correspondences for egomotion estimation. We use an available implementation [2] of this method for our evaluation. The table in the Figure 3 show some results of our experimentation. The comparisons shown are between the proposed method, Domke's method and the linear approach. For the proposed and linear approaches, point correspondences are obtained between images using SIFT based matching. For the Eight point approach, fundamental matrix is calculated based on this matches using RANSAC. Essential matrix obtained from the fundamental matrix and is decomposed to obtain the solution for egomotion. It can observed that the proposed approach outperforms both the linear and Domke's method.

In the case of real images, SIFT based feature matching was used to generate the point correspondences which are then used to calculate the JFD. To show the validity of our approach for real images we have calculated the epipolar lines based on the egomotion

[1] SOFA synthetic sequences courtesy of the Computer Vision Group, Heriot-Watt University (http://www.cee.hw.ac.uk/ mtc/sofa)

[2] http://www.cs.umd.edu/ domke/egomotion/



(a) (b)

Figure 4: Epipolar lines calculated based on the egomotion, estimated by proposed approach, overlaid over the real images in (a) and (b).

evaluated using the proposed approach. Figure 4 shows the epipolar lines overlaid on the respective images.

We also have experimented by varying the number of points in the equation (14) and have found that taking few equally spaced points on the image gives accurate results. As observed in Figure 5 increasing the number of equally spaced point beyond 36 (i.e. grid size= 6) does not yield a significant improvement in the accuracy of the estimated egomotion.

5 Conclusion and Discussions

In this paper, we have described a method to compute the egomotion of a moving camera using the statistical concept of joint feature distributions (JFD). The JFD's captured the statistics of a given collection of training correspondences and we used them to build a dense set of correspondences of the same kind. In this work, we focussed on using the JFD to predict correspondences and then used the epipolar constraint to find a probability distribution for the egomotion.

The JFD contains all the information regarding the uncertainties in egomotion. When the scene is decidedly deep, the fundamental matrix is well defined and we have a homogeneous covariance matrix for the family of epipolar lines associated with a given point x_i and the corresponding epipole e. So, the translation direction (e) and the rotation information (available from the fundamental matrix and e) can be calculated. For shallow scenes, the uncertainty in homographies is well characterized by the JFD information matrix (its inverse contains the homogeneous information of the homography) and this can be decomposed into translation and rotation components.

For scenes with varying degrees of projectivities (or collections of planes), a mixture of shallow JFD's (their shared eigenvector) would allow us to characterize the scene. We are currently exploring approaches that would allow us to compute the uncertainty in egomotion directly from this information without imposing the additional epipolar constraints.

References

[1] X. Armangu, H. Arajo, and J. Salvi. A review on egomotion by means of differential epipolar geomety applied to the movement of a mobile robot. *Pattern Recognition*, 21, 2003.

[2] W.F Clocksin. A new method for computing optical flow. In *Proceedings of British Machine Vision Conference*, 2000.

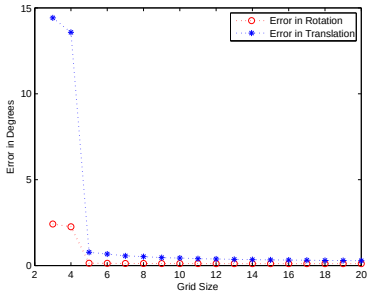


Figure 5: Improvement in accuracy of the estimated egomotion with increase in number of points under consideration in the proposed approach. n grid size means a equally spaced grid of $n \times n$ points on the image

[3] J. Domke and Y. Aloimonos. A probabilistic framework for correspondence and egomotion. In *ICCV Workshop on Dynamic Vision*, 2005.

[4] C.G Harris and M Stephens. A combined corner and edge detector. In *AVC*, 1988.

[5] Richard Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19, 1997.

[6] Allan D. Jepson and David J. Heeger. Linear subspace methods for recovering translational direction. In *Proceedings of the 1991 York conference on Spacial vision in humans and robots*, 1993.

[7] K. Kanatani. Unbiased estimation and statistical analysis of 3-d rigid motion from two views. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15, 1993.

[8] Kenichi Kanatani. 3-d interpretation of optical flow by renormalization. *Int. J. Comput. Vision*, 11, 1993.

[9] Hongdong Li and Richard Hartley. 5-point motion estimation made easy. In *Proceedings of the International Conference on Pattern Recognition*, 2006.

[10] David Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, 2004.

[11] David Nister. An efficient solution to the five-point relative pose problem. In *Proceedings of Computer Vision and Pattern Recognition*, 2003.

[12] K. Prazdny. Egomotion and relative depth map from optical flow. *Biological Cybernetics*, 36, 1980.

[13] Y Rosenberg and M Werman. Representing local motion as a probability distribution matrix applied to object tracking. In *Proceedings of Computer Vision and Pattern Recognition*, 1997.

[14] E.P Simoncelli, E.H Adelson, and D.J Heeger. Probability distributions of optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1991.

[15] Tina Y. Tian, Carlo Tomasi, and David J. Heeger. Comparison of approaches to egomotion computation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 1996.

[16] Bill Triggs. Factorization methods for projective structure and motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1996.

[17] Bill Triggs. Joint feature distributions for image correspondence. In *ICCV*, 2001.

[18] Zhengyou Zhang. Determining the epipolar geometry and its uncertainty: A review. Technical Report 2927, Sophia-Antipolis Cedex, France, 1996.

On New View Synthesis Using Multiview Stereo

O. J. Woodford[1], I. D. Reid[1], P. H. S. Torr[2] and A. W. Fitzgibbon[3]
[1]Department of Engineering Science, University of Oxford
[2]Department of Computing, Oxford Brookes University
[3]Microsoft Research, Cambridge

Abstract

We show that application of modern multiview stereo techniques to the new-view synthesis (NVS) problem introduces a number of non-trivial complexities. By simultaneously solving for the colour and depth of the new-view pixels we can eliminate the visual artefacts that conventional NVS-via-stereo suffers. The global occlusion reasoning which has led to considerable improvements in recent stereo algorithms can easily be included in the new algorithm, using a recently improved graph-cut-based optimizer for general multi-label conditional random fields (CRFs). However, the CRF priors that are important to success in stereo cannot be easily applied if the reconstruction is to be computed in the reference frame of the novel view. We address this problem by extending recent work on the fast optimization of texture priors in NVS to model the image edge structure, yielding a synthesis of the two approaches which yields good results on difficult image sequences.

1 Introduction

The problem addressed in this paper is new view synthesis (NVS): given multiple images of a 3D scene captured by a set of cameras, or by a single moving camera, generate a synthetic view of the scene, as it would appear from a new viewpoint. Such new views can be used in teleconferencing [1] or in 3-dimensionalizing monocular film footage.

Algorithms to solve this problem can be subdivided into two categories: scene reconstruction, and image-based rendering. Reconstruction methods form a representation of the 3D scene, for example as a 3D depth map [9], volumetric grid [11] or plenoptic function [7], from which the new view can be rendered. Stereo methods in particular can produce extremely accurate reconstructions, with only sparse input images, as occlusion between pixels is explicitly modelled [5, 12], and the smoothness prior can encourage depth discontinuities in the reconstruction to coincide with intensity edges in the input images—a conditional random field (CRF) prior [3]. However, the considerable, and universal, disadvantage of these methods is the generation of artefacts when the new view is finally rendered, such as "tearing" [9], distortion of fine features, and general aliasing caused by the change in reference frame.

In contrast, image-based rendering (IBR) methods solve directly for colour in the *new* view, thus avoiding these pitfalls. IBR methods can be further categorized into implicit and explicit geometry methods. Of these, implicit geometry methods [2, 13] marginalize out the depth, solving only for the colour of new-view pixels. Such methods generally employ image-based priors, working well on fine scene features. Explicit geometry methods [10] generate a depth map for the new view, much in the same way as traditional stereo

methods, but the important link between input image edges and depth discontinuities provided by the CRF is lost. IBR methods have also accounted for occlusion between pixels using occlusion models based on robust statistics [10, 13] rather than geometry, so do not enjoy the global occlusion reasoning of the stereo methods.

In this paper we combine these strands of stereo reconstruction and IBR. We take a recently introduced stereo algorithm [12] and adapt it to the NVS domain, requiring that a number of nontrivial problems be addressed. The primary contributions are (1) simultaneously solving for the new view and depth, with occlusion modelling, and (2) replacing the CRF with an efficient texture prior [13]. While stereo literature sometimes alludes to its potential application in NVS, the conversion process and the challenges it produces have not been addressed until now. This is, to our knowledge, the first IBR method to use a geometrical occlusion model in a global optimization framework, and is certainly the first to combine this with a texture term.

The paper proceeds in these stages: formal statement of the problem; definition of the energy function to be minimized; description of the graph-cut based optimization strategy; and evaluation of the results.

2 Problem statement

The task of NVS is to generate a new view, $\mathcal{V}$, of a scene, given a set of calibrated input views, $\mathcal{I}_1, .., \mathcal{I}_N$. A 2D vector, $\mathbf{x}$, denotes a pixel location in $\mathcal{V}$, the colour of which is written as $V(\mathbf{x})$. A projection function $\pi_i(\mathbf{x}, z)$ computes the 2D projection in image i of the 3D point at depth z in front of pixel $\mathbf{x}$ in the novel view. This function is easily computed from the images using commercial camera calibration software. The colour of this pixel projected into image $\mathcal{I}_i$ is written $I_i(\mathbf{x}, z)$, shorthand for $I_i\left(\pi_i(\mathbf{x}, Z(\mathbf{x}))\right)$, with $Z(\mathbf{x})$ (and z) being the estimated depth of the pixel. Pixel colours at non-integer locations are linearly interpolated from the image; locations outside the image boundaries are given a value of ∞.

The problem is poorly constrained—many candidate solutions $\mathcal{V}$ can explain the data equally well—so a powerful prior is needed to select good solutions. Following many current NVS [10] and stereo [5, 12] approaches, we cast our problem in a CRF energy minimization framework explicitly over depth (as well as colour), in contrast to methods which marginalize out depth, optimizing solely over colour [2, 13]. Our objective function contains costs over pixels and cliques of pixels, of the form

$$E(\mathcal{V}, \mathcal{Z}) = \underbrace{E_{\text{photo}}(\mathcal{V}, \mathcal{Z})}_{\text{data costs}} + \underbrace{E_{\text{smooth}}(\mathcal{V}, \mathcal{Z})}_{\text{surface smoothness}}, \tag{1}$$

for which we can use a powerful global optimizer based on graph cuts to compute strong local optima of the energy.

2.1 Data costs

The data cost is a term that ensures that each pixel in $\mathcal{V}$ is photo consistent with the input views. It enforces the constraint that the colour of output pixels which are visible (not occluded) in a given input view should match the colour of their projected location in that

view. We use a standard truncated SSD data cost. E_{photo} is the sum of data costs over all pixels in the novel view, denoted by the set $\mathcal{X}$, averaged over input views, thus:

$$E_{\text{photo}}(\mathcal{V},\mathcal{Z}) = \frac{1}{N}\sum_{i=1}^{N}\sum_{\mathbf{x}\in\mathcal{X}} O_i(\mathbf{x},\mathcal{Z})\min(\|V(\mathbf{x})-I_i(\mathbf{x},z)\|^2,\kappa)+(1-O_i(\mathbf{x},\mathcal{Z}))\nu \quad (2)$$

where κ is a robustness threshold, ν is a penalty cost for occluded pixels, and $O_i(\mathbf{x},\mathcal{Z})$ indicates whether pixel $\mathbf{x}$ is occluded in $\mathcal{I}_i$; 1 means visible, 0 means occluded. We must have $\nu > \kappa$ in order to avoid our objective function encouraging self-occlusions.

We use the asymmetrical occlusion model of Wei and Quan [12] to evaluate the visibility of pixels—the value of $O_i(\mathbf{x},\mathcal{Z})$ is determined entirely from our single depth map, $\mathcal{Z}$. It is defined to be 0 if there exists another pixel, $\mathbf{p}$, which projects to the same point[1] in $\mathcal{I}_i$ as pixel $\mathbf{x}$, and for which the projected depth is less than that of $\mathbf{x}$, otherwise it is 1.

2.2 Surface smoothness

Surface smoothness priors regularize out uncertainties in depth, especially in untextured regions, by placing a cost, $S()$, on a neighbourhood, $\mathcal{N}$, of pixels, which encourages smoothness. E_{smooth} is the sum of smoothness costs over a defined set of pixel neighbourhoods, $\mathbb{N}$, commonly defined as:

$$E_{\text{smooth}}(\mathcal{Z}) = \sum_{\mathcal{N}\in\mathbb{N}} \lambda_{\text{s}} \min\Big(S(\mathcal{N},\mathcal{Z}),\delta_{\text{s}}\Big) \quad (3)$$

where λ_{s} weights the smoothness prior, and δ_{s} is a discontinuity preserving threshold. This is a truncated linear kernel, which approaches the Potts model kernel as $\delta_{\text{s}} \to 0$.

Stereo methods in this graph cut optimized framework generally use, as a smoothness cost, a prior on the first order of disparity of two pixel neighbourhoods:

$$S(\{\mathbf{p},\mathbf{q}\},\mathcal{Z}) = \left|\frac{1}{Z(\mathbf{p})} - \frac{1}{Z(\mathbf{q})}\right|. \quad (4)$$

Many stereo methods locally vary λ_{s} and/or δ_{s} as a function of the reference image, in order to encourage occlusion boundaries to fit to image contours. Since, in NVS, the reference image (the new view, $\mathcal{V}$) is unknown, this approach is not possible here. However, Woodford *et al*. [13] recently introduced a pairwise texture prior which discourages discontinuities only where there is no supporting evidence from the input sequence. We therefore define a new E_{smooth} which incorporates this prior, thus:

$$E_{\text{smooth}}(\mathcal{V},\mathcal{Z}) = \sum_{\mathcal{N}\in\mathbb{N}} E_{\text{texture}}(\mathcal{V},\mathcal{N})\lambda_{\text{s}} \min\Big(S(\mathcal{N},\mathcal{Z}),\delta_{\text{s}}\Big), \quad (5)$$

$$E_{\text{texture}}(\mathcal{V},\mathcal{N}) = 1 + \lambda_{\text{t}} \min\Big(\min_{\mathbf{T}\in\mathbb{T}_{\mathcal{N}}} \|\mathbf{T}-\mathbf{V}(\mathcal{N},\mathcal{Z})\|^2,\delta_{\text{t}}\Big) \quad (6)$$

where $\mathbf{V}(\mathcal{N},\mathcal{Z})$ represents the vector of colours of the pixels in $\mathcal{N}$, defined by $\{V(x,\mathcal{Z})|x \in \mathcal{N}\}$, $\mathbb{T}_{\mathcal{N}}$ represents a library of patches specific to the $\mathcal{N}$, constructed as described in [13], and λ_{t} and δ_{t} are a further two model parameters.

[1] We define 'same point' to mean within half a pixel in both directions. This measure is an approximation, as different pixels have different projected footprints. While a more accurate definition could be employed, we found ours to work suitably well.

2.3 Computing colour

NVS differs from stereo in that one is optimizing over both colour and depth, as opposed to just depth. However, by making colour a function of depth, we can reuse the stereo optimization framework. We define the colour of pixel $\mathbf{x}$ to be the mean of visible input image samples of $\mathbf{x}$, thus:

$$V(\mathbf{x}, \mathcal{Z}) = \frac{\sum_{i=1}^{N} O_i(\mathbf{x}, \mathcal{Z}) I_i(\mathbf{x}, z)}{\sum_{i=1}^{N} O_i(\mathbf{x}, \mathcal{Z})}. \tag{7}$$

While the truncation term, κ, means that equation (2) is not necessarily unimodal in $\mathcal{V}$, given $\mathcal{Z}$, if we assume that all visible samples are a good match (as they should be for the correct solution), then equation (7) gives the colour that minimizes the E_{photo} term. Therefore, we can rewrite all the above energies in terms of $\mathcal{Z}$ only, and, by discretizing depth, we can now optimize this energy using a recently introduced method to obtain high-quality solutions, as we now describe.

3 Optimization

Despite the apparent complexity of the energy in fig 1, it ultimately boils down to an energy of the form

$$E(\mathcal{Z}) = \underbrace{\sum_{\mathbf{x} \in \mathcal{X}} u_{\mathbf{x}}(Z(\mathbf{x}))}_{\text{unary terms}} + \underbrace{\sum_{\mathcal{N} \in \mathbb{N}} c_{\mathcal{N}}(Z(\mathcal{N}))}_{\text{clique terms}} \tag{8}$$

where the cliques include 2-cliques of pixels which may be a long way apart, defining the occlusion term O (for which the reader is referred to [12]). A recent study of optimization algorithms [4] showed that such long range and irregularly connected terms are only effectively optimized using graph cut algorithms.

In order to optimize the energy, we therefore follow recent work [8], and reduce it to a sequence of binary problems as follows. Suppose we have a current estimate of the depth, $\mathcal{Z}_t$, and a *proposal* depth map $\mathcal{Z}_p$. The goal is to optimally combine ("fuse") the proposal and current depth maps to generate a new depth map $\mathcal{Z}_{t+1}$ for which the energy $E(\mathcal{Z}_{t+1})$ is lower than $\mathcal{Z}_t$. This is achieved by taking each pixel in $\mathcal{Z}_{t+1}$ from one of $\mathcal{Z}_t, \mathcal{Z}_p$, as controlled by a binary indicator image $\mathcal{B}$ with elements $B(\mathbf{x})$:

$$\mathcal{Z}(\mathcal{B}) = \mathcal{B} \cdot \mathcal{Z}_t + (1 - \mathcal{B}) \cdot \mathcal{Z}_p, \tag{9}$$

where dot indicates elementwise multiplication. Then the energy $E(\mathcal{Z})$ is a function only of the indicator image $\mathcal{B}$, so we may define

$$\mathcal{Z}_{t+1} = \mathcal{Z}\left(\operatorname*{argmin}_{\mathcal{B}} E(\mathcal{B} \cdot \mathcal{Z}_t + (1 - \mathcal{B}) \cdot \mathcal{Z}_p) \right). \tag{10}$$

If this binary optimization problem leads to a submodular[2] graph then a globally optimal $\mathcal{B}$ can be found using graph cuts. However, as Wei and Quan [12] explain, the occlusion term O is not guaranteed to fulfil the submodularity constraint.

[2] A submodular pairwise energy graph is one for which every pairwise energy term, $\phi_{pq}(l_p, l_q)$, $l_p, l_q \in \{0, 1\}$, satisfies the submodularity constraint: $\phi_{pq}(0,0) + \phi_{pq}(1,1) \leq \phi_{pq}(0,1) + \phi_{pq}(1,0)$.

$$
\begin{aligned}
V(\mathbf{x}, \mathcal{Z}) &:= \frac{\sum_{i=1}^{N} O_i(\mathbf{x}, \mathcal{Z}) I_i(\mathbf{x}, z)}{\sum_{i=1}^{N} O_i(\mathbf{x}, \mathcal{Z})} \\
E_{\text{photo}}(\mathcal{Z}) &:= \frac{1}{N} \sum_{i=1}^{N} \sum_{\mathbf{x} \in \mathcal{X}} \begin{pmatrix} O_i(\mathbf{x}, \mathcal{Z}) \min(\|V(\mathbf{x}, \mathcal{Z}) - I_i(\mathbf{x}, z)\|^2, \kappa) \\ + \quad (1 - O_i(\mathbf{x}, \mathcal{Z}))\, \nu \end{pmatrix} \\
S(\{\mathbf{p}, \mathbf{q}\}, \mathcal{Z}) &:= \left| \frac{1}{Z(\mathbf{p})} - \frac{1}{Z(\mathbf{q})} \right| \\
E_{\text{texture}}(\mathcal{V}, \mathcal{N}) &:= 1 + \lambda_{\text{t}} \min\left(\min_{\mathbf{T} \in \mathbb{T}_{\mathcal{N}}} \|\mathbf{T} - \mathbf{V}(\mathcal{N}, \mathcal{Z})\|^2, \delta_{\text{t}} \right) \\
E_{\text{smooth}}(\mathcal{Z}) &:= \sum_{\mathcal{N} \in \mathbb{N}} E_{\text{texture}}(\mathcal{V}, \mathcal{N}) \lambda_{\text{s}} \min\left(S(\mathcal{N}, \mathcal{Z}), \delta_{\text{s}} \right) \\
E(\mathcal{Z}) &:= \underbrace{E_{\text{photo}}(\mathcal{Z})}_{\text{data costs}} + \underbrace{E_{\text{smooth}}(\mathcal{Z})}_{\text{surface smoothness}}
\end{aligned}
$$

Figure 1: **Energy function.** The energy $E(\mathcal{Z})$ minimized as a function of the new-view depth map $\mathcal{Z}$. Note that although complex, with many terms, this function can be effectively reduced to a sequence of binary optimization problems, for which the QPBO algorithm finds either a global optimum, or a local optimum with an indication of how far from the global optimum it is.

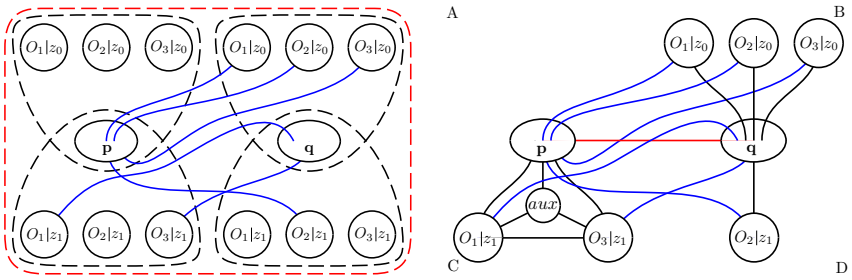
Rather, we can now use the Quadratic Pseudo-Boolean Optimization (QPBO) strategy introduced to computer vision in [8]. QPBO is an extension of graph cuts that can be used to optimize non-submodular energies. Unlike the globally optimal submodular case, QPBO returns a partial solution to $\mathcal{B}$ and an associated mask $\mathcal{M}$, with the guarantee that at pixels $\mathbf{x}$ where $M(\mathbf{x}) = 1$, the value $B(\mathbf{x})$ is at the value it would have at the global minimum, but pixels where $M(\mathbf{x}) = 0$ have "unlabelled" values. A further guarantee of QPBO is that, after forcing $B(\mathbf{x}) = 1$ at those unlabelled pixels, $E(Z_{t+1}) \leq E(Z_t)$, thus ensuring a convergent optimization. In practice, we find that, while there may be many unlabelled pixels at each fusion step, those pixels for which the proposal depth is optimal tend to be labelled, so the energy is minimized quite effectively.

In principle our choice of proposal depth map is not constrained when using QPBO, but we emulate the simple approach of [12] in setting the proposal at each fusion step to be a fronto-parallel plane at one of a discrete set of depths.

3.1 Graph construction

NVS has the additional complexity over stereo that the colour of pixel $\mathbf{x}$, as given by equation (7) depends not only on its depth (current or proposed), but also on the binary visibilities $O_1(\mathbf{x}, \mathcal{Z}), .., O_N(\mathbf{x}, \mathcal{Z})$. Therefore, in order to accurately model the energy of equation (2) our graph requires cliques of size $N + 1$, as shown in figure 2(a), while equation (6) requires cliques of size $4N + 2$.

QPBO, like all graph cut algorithms, can only solve graphs with cliques up to size two. Energy terms of any order can always be decomposed into a set of pairwise energy terms, with additional, latent nodes, but this set grows exponentially with the clique size. In order to avoid an explosion in graph complexity, we limit our maximum clique size to three. This requires approximations to be made in our graph structure, the details of



(a) Exact energy representation. (b) Approximate, soluble construction.

Figure 2: **Graph construction**. Graphical representations of (a) our objective function, and (b) the approximate energy graph we construct, for a 2×1 pixel image, with $N = 3$. Ellipses (including circles) represent nodes of the graph (and associated unary terms); lines (edges) represent pairwise energy terms. Nodes $\mathbf{p}$ and $\mathbf{q}$ encode the depth labels of the two image pixels. The nodes $O_1|z_0$, *etc*. encode whether (by way of example) pixel $\mathbf{p}$ is occluded at depth label 0 (i.e. depth z_0) in $\mathcal{I}_1$, given the depth labels of all other pixels also. The blue lines are infinite edge costs which set these visibilities, as per [12]. The list of occlusion interactions (*i.e*. blue lines) is computed prior to solving the graph, and it should be noted that not every occlusion node has such an interaction, while others may have more than one. The dashed lines in (a) encircle nodes in higher order cliques, which accurately model the data costs (black lines) of equation (2), and surface smoothness cost (red line) of equation (5). However, since graph cut optimizers can only solve graphs with pairwise and unary terms, we approximate these cliques to generate graph (b) as follows. First, we approximate the surface smoothness cost with a single pairwise edge (red line), by using a fixed approximation of pixel colour in equation (6). Then we remove all occlusion nodes with no occlusion interactions—the image samples associated with those nodes will always be visible—reducing some of the cliques in size. Cliques of size 1, 2 and 3 can then be modelled exactly using unary and pairwise terms (black lines), as shown by the graph structures in corners A, D and C of (b) respectively. In particular, the triple clique energy is decomposed into 6 pairwise terms according to [6], which also generates an additional, latent node, aux. Cliques of size 4 (corner B) or larger are approximated using a set of pairwise edges, as described in §3.1.

which are one of the main contributions of this paper.

To remove the complexity generated by the variability of colour in equation (6), we simply fix the colour of each pixel $\mathbf{x}$ at a given depth z to $V'(\mathbf{x}, \mathcal{Z}_t, z)$, *i.e*. we assume all pixels other than $\mathbf{x}$ to be at the depth output by the previous fusion, thus:

$$V'(\mathbf{x}, \mathcal{Z}_t, z) = \frac{\sum_{i=1}^{N} O_i(\mathbf{x}, \mathcal{Z}_t) I_i(\mathbf{x}, z)}{\sum_{i=1}^{N} O_i(\mathbf{x}, \mathcal{Z}_t)}. \tag{11}$$

Rather than use this approximation as standard in equation (2) as well, we prefer to model the data costs as accurately as possible, as they have a much greater impact on the quality of the solution. Figure 2(b) shows that, once unnecessary occlusion nodes have been removed from the graph, pixels at a given depth with up to two possibly occluded input samples can be modelled exactly with a single unary, pairwise or triple clique term for the data cost over all input images. We can therefore model all data costs exactly when $N = 2$. However, in the case of larger cliques we use the fixed colour, V', in evaluating E_{photo}. Potentially occluded image samples therefore generate a pairwise edge, as in stereo [5, 12], while data costs for unoccluded samples are simply added to the correct unary term of the node representing the pixel in question.

The approximation of equation (11) means we no longer model the true value of our objective function in our graph. When we evaluate the true value of colour, $V(\mathbf{x}, \mathcal{Z}_{t+1})$, given by equation (7), after the fusion operation, some of the pixels will change colour due to the visibilities of the input image samples changing with the new depth map. The result is that the objective energy $E(\mathcal{Z}_{t+1})$ may increase, such that the guarantee of convergence given by the stereo framework is lost. However, we have found this to be rare and negligible in practice.

4 Experiments

In all our experiments we use the parameter set given in table 1, which we chose after a grid search over parameter space and qualitative inspection of the results. We make two passes through the set of depth proposals, which is dependent on the sequence, but numbers of the order of 100 depths spaced equally in disparity space (1/depth); the passes run through the set in order, from near to far. The first pass fixes most pixels, with the second making only a few corrections. While additional passes improve the result further, returns on computation time diminish rapidly. We ran experiments on a range of standard NVS and stereo image sequences, and compared our results with other methods.

Figures 3 & 4 show images synthesized from a viewpoint halfway between the two rectified input views. The former compares our method with warping a known view with a depth map [9] (here we use ground truth[3]). Warped stereo leaves holes sometimes (cyan pixels), but also sets a single depth for mixed depth pixels, which then causes artefacts (*e.g*. around depth discontinuities) when rendered in the new view. By rendering directly into the new view we avoid this rerendering step and its associated artefacts.

Figure 3 also demonstrates the impacts on our synthesis framework, our main contribution, of two further contributions of our work—employing a texture prior to weight the surface smoothness cost, and sensibly approximating data costs in our graph. In image (d) (no texture prior), some of the cone tips are truncated. The aim of our texture prior is to encourage depth discontinuities to fit to the edges of objects, and we can see in (c) that these cone tips have been corrected, as desired. Comparing (c) with (e) demonstrates that accurately modelling data costs in cliques with less than three potentially occluded pixels produces far fewer rendering artefacts, though this improvement becomes less pronounced as N increases.

Figure 4 shows a comparison of our method with the DP method of Criminisi *et al*. [1]. While producing similar results on this sequence, and in real-time, their method enforces the less general "ordering" constraint in modelling occlusions. Our approach is therefore preferable in scenes with complex foreground objects and wide baseline input views. Note that the input images have different exposures—this is handled by equalizing the mean and variance of the two images.

Figure 5 shows a new view of a challenging sequence, with many occlusions, synthesized from 8 input images. Our method is able to reconstruct the colour in occluded regions (*e.g*. wall above nose, and between ribs) well, in contrast to the implicit depth method of Woodford *et al*. The explicit depth model and smoothness prior allows us to extract the correct depth of the wall, and the geometric occlusion model the correct

[3] Sequence and ground truth depth maps downloaded from www.middlebury.edu/stereo.

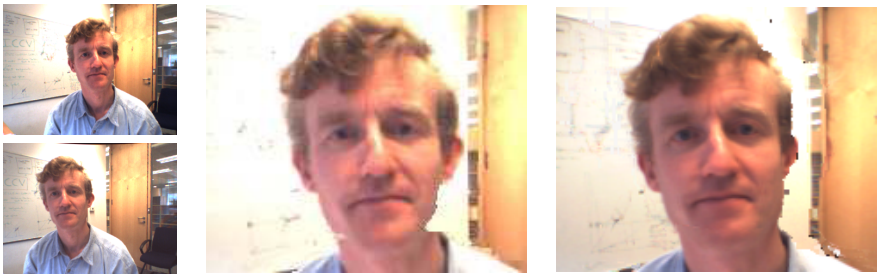
Parameter	κ	ν	δ_s	λ_s	δ_t	λ_t
Value	$c(12.5N/(N-1))^2$	$\kappa+1$	$1.9d$	$0.24\kappa/\delta_s$	$5000c$	$6/\delta_t$

Table 1: **Parameter settings**. Values of the constant parameters in our objective function, where c is the number of colour channels in the input sequence, and d is the constant disparity spacing between the discrete proposal depths, which varies between input sequences.

(a) Ground truth (b) Warped stereo views

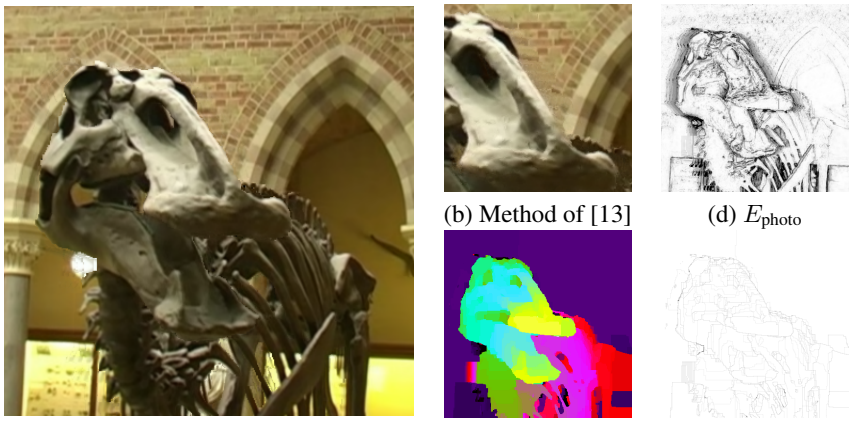
(c) Our result (d) $\lambda_t = 0$ (e) Always fix to V'

Figure 3: **Cones sequence**. (a) A ground truth central view, and (b) a view synthesized by warping (in a manner similar to that of [9]) two outer images into the central view using ground truth depth maps—blue pixels are unknown due to holes in the depth maps, while cyan pixels are regions occluded in both input views. Our result (c), and our results (d) removing the texture prior and (e) using the approximate colour of equation (11) in all data cost calculations.



(a) Input views (b) Result of [1] (b) Our result

Figure 4: **Teleconferencing**. Rendering a centre view (c) from 2 rectified input views, for direct gaze teleconferencing. Sequence taken from [1], with the result from the same paper (b).



(a) Our result (b) Method of [13] (c) $1/\mathcal{Z}$ (d) E_{photo} (e) E_{smooth}

Figure 5: **Edmontosaurus sequence**. (a) New view of a sequence from [13], and the result of the method of the same paper (b). (c)–(e) show other outputs of our method, as labelled. $N = 8$.

texture. Some artefacts, such as shadows and jaggedness, exist around the edges of the foreground object.

Figure 6 demonstrates the results of our algorithm on a further two difficult sequences. While fine details such as fur and feathers are accurately rendered, some areas (*e.g.* under the forearm and upper arm in (a), to left of head in (b)) appear blurred; this is due to the wrong depth being chosen in these regions.

Artefacts in our results are generated by a combination of two processes: (1) the optimal solution to our objective function not accurately representing the scene, and (2) nodes being unlabelled in each fusion step when the optimal solution would select the proposed new depth. We found that optimizing parameters for a particular sequence or view often produced better results than with the standard parameter set—future work may involving developing methods to automatically evaluate the optimal settings. We expect the performance of QPBO, an algorithm relatively new to the field, to improve significantly in the future, further reducing the appearance of artefacts.

(a) Our result (b) Our result (c) Ground truth

Figure 6: **Monkey and plant & toy sequences**. (a) A new view of the monkey sequence (from [2]). $N = 8$. (b) A leave-one-out test on the plant & toy sequence (from [13]). (c) The ground truth view of (b). $N = 8$.

5 Conclusion

We have confirmed the common suggestion that graph-cut stereo methods can be applied to the task of new-view synthesis. While straightforward in principle, this repurposing presents a number of technical difficulties, the solutions to which are the main contributions of this paper. The results improve on the current state of the art NVS methods, demonstrating the power of an explicit depth model with global, geometric occlusion reasoning in determining colour in partially occluded regions, as well as showing that rendering directly into the new view avoids artefacts generated by scene reconstruction methods. While the texture prior which we apply is not in principle as powerful as the stereo CRF prior (which cannot be applied), we show that it acts similarly in improving rendering at discontinuity boundaries.

Acknowledgements We are extremely grateful to Carsten Rother and Vladimir Kolmogorov for providing us with a preliminary version of [8], QPBO software and for generous assistance in using the latter. We also thank Vladimir for discussing graph cut stereo with us. Research funded by EPSRC and Sharp.

References

[1] A. Criminisi, J. Shotton, A. Blake, C. Rother, and P. H. S. Torr. Efficient dense stereo with occlusions for new view-synthesis by four-state dynamic programming. *IJCV*, 71(1):89–110, Jan 2007.

[2] A. Fitzgibbon, Y. Wexler, and A. Zisserman. Image-based rendering using image-based priors. In *Proc. ICCV*, volume 2, pages 1176–1183, Oct 2003.

[3] V. Kolmogorov, A. Criminisi, A. Blake, C. Cross, and C. Rother. Probabilistic fusion of stereo with color and contrast for bi-layer segmentation. *IEEE PAMI*, 28(9):1480–1492, Sep 2006.

[4] V. Kolmogorov and C. Rother. Comparison of energy minimization algorithms for highly connected graphs. In *Proc. ECCV*, volume 2, pages 1–15, 2006.

[5] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *Proc. ECCV*, volume 3, page 82, 2002.

[6] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE PAMI*, 26(2):147–159, 2004.

[7] M. Levoy and P. Hanrahan. Light field rendering. In *SIGGRAPH96*, 1996.

[8] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer. Optimizing binary MRFs via extended roof duality. In *Proc. CVPR*, 2007.

[9] D. Scharstein. Stereo vision for view synthesis. In *Proc. CVPR*, pages 852–858, 1996.

[10] C. Strecha, R. Fransens, and L. Van Gool. Wide-baseline stereo from multiple views: a probabilistic account. In *Proc. CVPR*, volume 1, pages 552–559, Jun 2004.

[11] G. Vogiatzis, P. H. S. Torr, and R. Cipolla. Multi-view stereo via volumetric graph-cuts. In *Proc. CVPR*, pages 391–398, 2005.

[12] Y. Wei and L. Quan. Asymmetrical occlusion handling using graph cut for multi-view stereo. In *Proc. CVPR*, volume 2, pages 902–909, 2005.

[13] O. J. Woodford, I. D. Reid, and A. W. Fitzgibbon. Efficient new view synthesis using pairwise dictionary priors. In *Proc. CVPR*, 2007.

Author Index

Keyword Index